

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



Máster Universitario en Investigación e  
Innovación en Tecnologías de la Información  
y las Comunicaciones ( $i^2$ -TIC)

TRABAJO FIN DE MÁSTER

# ALGORITMOS PARA EL ESTABLECIMIENTO DE CONEXIONES NEURONALES HÍBRIDAS

Autor: Manuel Reyes Sánchez  
Director: Pablo Varona Martínez

Junio 2017



# ALGORITMOS PARA EL ESTABLECIMIENTO DE CONEXIONES NEURONALES HÍBRIDAS

Autor: Manuel Reyes Sánchez  
Director: Pablo Varona Martínez

Grupo de Neurocomputación Biológica (GNB)  
Dpto. de Ingeniería Informática  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
Junio 2017





# Resumen

Un circuito neuronal híbrido es aquel formado por neuronas biológicas vivas y neuronas artificiales. Estos circuitos tienen múltiples utilidades y pueden ser empleados para estudiar el funcionamiento y la dinámica de la actividad neuronal. También puede tener futuros usos clínicos, permitiendo sustituir elementos dañados o reparar dinámicas del circuito.

Las neuronas artificiales pueden ser implementadas mediante hardware o mediante su simulación software. En ambos casos debe tenerse en cuenta la necesaria adaptación de las señales para que puedan trabajar con las neuronas biológicas en interacción bidireccional. Por ejemplo, se deben escalar la amplitud de entrada/salida o la temporalidad del sistema. Por otro lado, también es necesario adaptar las neuronas artificiales para comportarse de determinada manera según el sistema vivo al cual se estén conectando.

Ante esta necesidad de adaptación dinámica, la simulación software ofrece múltiples ventajas frente a las implementaciones hardware, gracias a su mayor flexibilidad y adaptabilidad. Esto permite el análisis y el control online del circuito híbrido, con lo que es posible realizar procesos de estimulación en ciclo cerrado dependientes de la actividad observada.

Por lo anterior, en este trabajo se propone un conjunto de protocolos para la construcción, acondicionamiento y adaptación de circuitos híbridos para que puedan ser implementados de manera sencilla y eficaz. Además, se ha realizado una implementación de los protocolos propuestos utilizando algoritmos y calibraciones concretas. Los algoritmos incluyen métodos para asimilar las señales de los modelos a la señal neuronal viva de manera automática, así como procesos de ciclo cerrado para establecer los parámetros que maximicen un objetivo de conexión cuantificable. Para la implementación ha sido necesaria la utilización de un sistema operativo de tiempo real con el objetivo de cumplir las restricciones de décimas de milisegundos que impone la actividad neuronal en muchos tipos de circuitos híbridos.

Los protocolos definidos pueden ser también utilizados para realizar una automatización de experimentos y de esta manera realizarlos con una optimización de la esperanza de vida de la preparación, permitiendo, por ejemplo, la exploración de dinámicas para diferentes condiciones del sistema, o el control eficaz de la actividad neuronal.

Finalmente, los algoritmos y el programa realizado se han valido construyendo circuitos híbridos con motoneuronas del ganglio estomatogástrico del cangrejo de mar común (*Carcinus maenas*). La señal de los modelos ha sido adaptada dinámicamente a los rangos biológicos de cada experimento y los procesos en ciclo cerrado han permitido ajustar el comportamiento de los modelos de neuronas y sinapsis, así como automatizar experimentos, e implementar exploraciones dinámicas.

## Palabras Clave

Circuitos híbridos, calibración automática, modelos neuronales, neurociencia en ciclo cerrado, sinapsis artificial, Linux, Comedi, software de tiempo real, pinzamiento dinámico, electrofisiología, CPG, automatización de experimentos, exploración dinámica.



# Abstract

Hybrid neuron circuits are formed by biological living neurons and artificial neurons. These circuits have multiple applications and can be used to understand the dynamics of neural activity. They can also have future clinical applications, allowing the replacement of damaged elements or the repairmen of circuit dynamics.

Artificial neurons can be built through hardware implementation or through software simulation. In both cases the adaptation of the signals must be taken into account so they can work together with biological neurons in bidirectional interaction. For example, the input/output amplitude or the temporality of the system must be scaled. On the other hand, it is also necessary to adapt the artificial neurons to a specific behavior according to the living system to which they are connected.

Due to the need of dynamic adaptation, software simulations offer many advantages (in relation to hardware solutions), because of their flexibility and adaptability. This allows online analysis of the results. It also enables closed-loop activity dependent stimulation.

Therefore, this work proposes a set of protocols to build and adapt hybrid circuits so that they can be implemented in a simple straightforward manner. In addition, an implementation of the proposed protocols has been made using specific algorithms and calibrations. These algorithms include methods to automatically scale signals from the models to the levels observed in living neurons, as well as closed-loop processes to establish parameters that maximize a quantitative connection target. For the implementation, a real-time operating system was need in order to meet the time constrains in the range of tenths of milliseconds imposed by the neuronal activity in many hybrid circuits.

The defined protocols can be useful to implements an automatic calibration of the experiments in a very precise way allowing, for example, the exploration of the behavior of different conditions of the system, or the effective control of the neuronal activity.

Finally, the algorithms and programs have been validated by building hybrid circuits with motoneurons of the stomatogastric ganglion of the shore crab (*Carcinus maenas*), successfully making these connections. The signal of the models has been correctly adapted to the biological ranges of each experiment and the closed-loop processes have allowed to adjust the behavior of the neuron and the synapse models, to automate experiments and to perform dynamic explorations.

## Key words

Hybrid circuits, automatic calibration, neuronal models, closed-loop neuroscience, artificial synapse, Linux, Comedi, real time software, dynamic clamp, electrophysiology, CPG, experiment automation, dynamic exploration.



# Agradecimientos

Por estar siempre ahí y por todo su apoyo no podría comenzar estos agradecimientos sin nombrar en primer lugar a mis padres, espero que podáis estar orgullosos de mí. También a mis abuelos por todos los valores que reflejan.

Quiero dar las gracias a todos mis amigos de Tres Cantos y de la UAM por su apoyo y amistad, por ser la brisa de aire fresco que de vez en cuando me salva de la extenuación. Y a Alex por escucharme cuando lo he necesitado.

Agradecer a mis compañeros del GNB que nuestro laboratorio sea un lugar tan bueno para trabajar (y perder un poco el tiempo) y que siempre podamos ayudarnos entre todos. Gracias Ángel, Carlos, Irene, Roy, Mario, Ángel, Carlos, Nerea, Aarón y Vinicio. Gracias a Paco por sus consejos.

Por último gracias a mi tutor, Pablo, que además de su ayuda y guía me ha descubierto un campo apasionante. Y por su interés en que pueda seguir aprendiendo con él, no tengo palabras para agradecerse.

*“We can only see a short distance ahead, but we can see plenty there that needs to be done.”*

Alan Turing



# Índice general

|  |             |
|--|-------------|
| <b>Índice de Figuras</b>   | <b>XIV</b>  |
| <b>Índice de Tablas</b>  | <b>XV</b>   |
| <b>Índice de Códigos</b>   | <b>XVII</b> |
| <b>Glosario</b>  | <b>XIX</b>  |
| <b>1. Introducción</b>   | <b>1</b>    |
| 1.1. Motivación . . . . .  | 1           |
| 1.2. Objetivos . . . . .   | 2           |
| 1.3. Estructura de la memoria . . . . .                                  | 3           |
| <b>2. Estado del arte</b>  | <b>5</b>    |
| 2.1. El sistema nervioso . . . . .                                       | 5           |
| 2.1.1. La neurona . . . . .  | 5           |
| 2.1.2. Sinapsis neuronal . . . . .                                       | 6           |
| 2.2. Utilidades del estudio de la actividad neuronal . . . . .           | 6           |
| 2.3. Observación de la actividad neuronal . . . . .                      | 7           |
| 2.4. Simulación de la actividad neuronal . . . . .                       | 8           |
| 2.5. Interacción en ciclo cerrado . . . . .                              | 10          |
| 2.6. Circuitos híbridos . . . . .  | 10          |
| 2.6.1. Esquema general de un circuito híbrido . . . . .                  | 11          |
| 2.7. Circuitos Generadores Centrales de Patrones . . . . .               | 12          |
| 2.8. Control de la actividad y exploración dinámica automática . . . . . | 13          |
| 2.8.1. Ejemplos de control y calibración automática . . . . .            | 13          |
| 2.8.2. Accesibilidad de la calibración automática . . . . .              | 14          |
| 2.9. Sistemas operativos de tiempo real . . . . .                        | 15          |
| <b>3. Hardware, software y preparación biológica</b>                     | <b>17</b>   |
| 3.1. Hardware . . . . .  | 17          |
| 3.1.1. Ordenadores . . . . .   | 17          |

|   |           |
|---|-----------|
| 3.1.2. Tarjetas de adquisición de datos . . . . .                                 | 18        |
| 3.1.3. Neuronas electrónicas . . . . .  | 18        |
| 3.2. Software . . . . .   | 19        |
| 3.3. Neuronas vivas . . . . .   | 19        |
| <b>4. Diseño</b>  | <b>23</b> |
| 4.1. Planteamiento inicial . . . . .  | 23        |
| 4.2. Esquema propuesto . . . . .  | 24        |
| 4.3. Requisitos . . . . .   | 26        |
| <b>5. Desarrollo</b>  | <b>29</b> |
| 5.1. Simulación de la dinámica neuronal . . . . .                                 | 29        |
| 5.1.1. Modelos neuronales . . . . .   | 29        |
| 5.1.2. Modelos de sinapsis . . . . .  | 31        |
| 5.2. Primer subsistema . . . . .  | 32        |
| 5.2.1. Análisis de las señales . . . . .  | 32        |
| 5.2.2. Cálculo del escalado temporal y de potencial . . . . .                     | 35        |
| 5.3. Segundo subsistema . . . . .   | 35        |
| 5.3.1. Integración del modelo neuronal . . . . .                                  | 35        |
| 5.3.2. Análisis de las señales y medida de la sincronización . . . . .            | 35        |
| 5.3.3. Ajuste dinámico de parámetros . . . . .                                    | 36        |
| 5.3.4. Frecuencia de interacción constante e intercambio de información . . . . . | 36        |
| <b>6. Experimentos y resultados</b>   | <b>39</b> |
| 6.1. Análisis y detección de eventos en la señal . . . . .                        | 39        |
| 6.1.1. Filtrado de la señal . . . . .   | 40        |
| 6.1.2. Detección de eventos . . . . .   | 41        |
| 6.2. Calibración de escalas de tiempo y de amplitud . . . . .                     | 42        |
| 6.3. Sinapsis artificial de dos neuronas externas al programa . . . . .           | 44        |
| 6.4. Sinapsis eléctrica . . . . .   | 45        |
| 6.5. Sinapsis química . . . . .   | 46        |
| 6.6. Calibrado de la conexión en ciclo cerrado . . . . .                          | 48        |
| 6.6.1. Calibración de la conductancia eléctrica . . . . .                         | 48        |
| 6.6.2. Calibración del comportamiento de los modelos . . . . .                    | 48        |
| 6.7. Resumen de los procedimientos realizados . . . . .                           | 48        |
| 6.8. Mapa de conductancias químicas . . . . .                                     | 52        |



|  |           |
|--|-----------|
| <b>7. Conclusiones y trabajo futuro</b>  | <b>53</b> |
| 7.1. Conclusiones . . . . .  | 53        |
| 7.2. Trabajo futuro . . . . .  | 54        |
| <b>Bibliografía</b>  | <b>59</b> |
| <b>Anexos</b>  | <b>61</b> |
| <b>A. Modelos neuronales y de sinapsis</b>                                       | <b>63</b> |
| A.1. Integración de ecuaciones diferenciales ordinarias en tiempo real . . . . . | 63        |
| A.1.1. Método de Euler . . . . .   | 63        |
| A.1.2. Método de Runge-Kutta . . . . .   | 64        |
| A.2. Interpolación lineal . . . . .  | 64        |
| A.3. Modelos neuronales . . . . .  | 64        |
| A.3.1. Hindmarsh-Rose . . . . .  | 64        |
| A.3.2. Izhikevich . . . . .  | 65        |
| A.3.3. Mapa de Rulkov . . . . .  | 66        |
| A.4. Modelos de sinapsis . . . . .   | 67        |
| A.4.1. Sinapsis eléctrica . . . . .  | 67        |
| A.4.2. Sinapsis química gradual . . . . .  | 67        |
| A.4.3. Sinapsis química dinámica . . . . .                                       | 68        |
| <b>B. Clamp-cli</b>  | <b>69</b> |
| B.1. Organización del proyecto . . . . .   | 69        |
| B.2. Instalación . . . . .   | 70        |
| B.2.1. Guía para Debian . . . . .  | 70        |
| B.3. Utilización . . . . .   | 71        |
| B.4. Fichero de datos . . . . .  | 72        |
| B.4.1. Fichero 1 . . . . .   | 72        |
| B.4.2. Fichero 2 . . . . .   | 73        |
| B.5. Representación gráfica . . . . .  | 73        |
| <b>C. Estudio de latencias</b>   | <b>75</b> |
| <b>D. Difusión del trabajo</b>   | <b>77</b> |



## Índice de Figuras

|  |    |
|--|----|
| 2.1. Circuito que simula el comportamiento pasivo de una membrana neuronal con tres canales. . . . .   | 9  |
| 2.2. Esquema de construcción de una interacción en ciclo cerrado y un circuito híbrido. . . . .  | 11 |
| 3.1. Señal producida por una neurona electrónica del GNB. . . . .  | 18 |
| 3.2. Fotografía del setup de electrofisiología del Grupo de Neurocomputación Biológica (GNB). . . . .  | 19 |
| 3.3. Fotografía de una neurona electrónica. . . . .  | 20 |
| 3.4. Cangrejo de mar común ( <i>Carcinus maenas</i> ) y detalle de la primera parte de su disección (estómago abierto) para los experimentos realizados. . . . .   | 20 |
| 3.5. Ganglio estomatogástrico con electrodos conectados a una neurona para registrar la actividad neuronal e inyectar corriente. . . . .   | 20 |
| 3.6. Imagen del microscopio tras un experimento. Se indican los elementos observados. . . . .  | 21 |
| 4.1. Esquema que representa el periodo constante de interacción con el sistema biológico y el tiempo disponible para realizar cálculos entre cada interacción a frecuencias de adquisición/envío de 10-20 kHz. . . . . | 23 |
| 4.2. Primer subsistema del diseño planteado. . . . .   | 24 |
| 4.3. Segundo subsistema del diseño planteado. . . . .  | 25 |
| 5.1. Ejemplos de los potenciales de membrana producidos por los modelos neuronales y por una neurona viva del CPG pilórico de un cangrejo. . . . .   | 30 |
| 5.2. Esquema de una sinapsis eléctrica. . . . .  | 31 |
| 5.3. Ilustración de una sinapsis dinámica. Figura adaptada de (Tsodyks and Markram, 1997). Se observa la reducción en la efectividad de la sinapsis en la neurona post sináptica. . . . .                              | 32 |
| 5.4. Diferencia en el número de puntos para generar una ráfaga entre dos modelos diferentes. . . . .   | 33 |
| 5.5. Ejemplo de diferencia de escala entre un modelo neuronal y la señal biológica. . . . .  | 34 |
| 6.1. Filtrado online de la señal neuronal adquirida. . . . .   | 40 |
| 6.2. Detalle del filtrado online de la señal neuronal adquirida. . . . .   | 40 |
| 6.3. Detección del primer disparo de una ráfaga neuronal . . . . .   | 42 |

|  |    |
|--|----|
| 6.4. Señal neuronal en origen con distinta escala de voltaje y temporalidad adaptada a la dimensión de una neurona electrónica del GNB. Unidades de voltaje Izhikevich = $[-74, 25; 30, 24]$ . Unidades adaptadas de voltaje Izhikevich = $[-3, 5; 6, 9]$ . Puntos por ráfaga Izhikevich = 59.324. Puntos por ráfaga adaptados Izhikevich = 2.500. . . . . | 43 |
| 6.5. Neuronas viva y electrónica sin interacción. Hay diferencia en la frecuencia de disparo. . . . .  | 44 |
| 6.6. Neuronas viva y electrónica conectadas por una sinapsis eléctrica. . . . .  | 44 |
| 6.7. Neuronas viva y electrónica conectadas en antifase por una sinapsis eléctrica. . .  | 45 |
| 6.8. Modelos neuronales conectados por sinapsis eléctricas a una neurona viva. Ejemplo de escala (modelo Hindmarsh-Rose): Voltajes originales = $[-1, 6; 1, 8]$ . Voltajes escalados = $[-5, 68; -3, 68]$ . Puntos ráfaga = 260.166. Puntos ráfaga escalado = 5.000. Conductancia $g=0.3$ . . . . .  | 45 |
| 6.9. Relación entre señales neuronales en una sinapsis eléctrica (Rulkov-LP). . . . .  | 46 |
| 6.10. Relación entre señales neuronales en una sinapsis eléctrica (Izhikevich-LP). . . . .   | 46 |
| 6.11. Modelos neuronales conectados por sinapsis químicas a una neurona viva. Ejemplo de escala (mapa de Rulkov): Voltajes originales = $[-1, 9; 2, 1]$ . Voltajes escalados = $[-6, 2; -3, 2]$ . Puntos ráfaga = 350. Puntos ráfaga escalado = 5.000. Conductancias: $g_{lenta}=0.02$ , $g_{rápida}=0.2$ . . . . .  | 47 |
| 6.12. Relación entre señales neuronales en una sinapsis química. . . . .   | 47 |
| 6.13. Corrientes en una sinapsis químicas. . . . .   | 47 |
| 6.14. Calibración automática de la conductancia de una sinapsis eléctrica por reducción del ECM (40 %). Modelo Izhikevich y neurona viva LP (a). Conductancia final $g = 0,2$ . . . . .  | 49 |
| 6.15. Calibración automática de la conductancia de una sinapsis eléctrica por reducción del ECM (40 %). Mapa de Rulkov y neurona viva LP (b). Conductancia final $g = 0,3$ . . . . .   | 49 |
| 6.16. Variación del parámetro $\mu$ del modelo Hindmarsh-Rose. Conexión con una neurona electrónica. Valor óptimo $\mu = 0,0131$ . . . . .   | 50 |
| 6.17. Variación del parámetro $\mu$ del modelo Hindmarsh-Rose. Conexión con una neurona LP viva. Valor óptimo $\mu = 0,0371$ . . . . .   | 50 |
| 6.18. Resumen del proceso para la realización de un circuito híbrido. . . . .  | 51 |
| 6.19. Fase generada por diferentes conductancias en una sinapsis química. . . . .  | 52 |
| A.1. Mapa de selección de los parámetros del modelo de Izhikevich. Figura reproducida de <a href="http://www.izhikevich.org/publications/spikes.htm">www.izhikevich.org/publications/spikes.htm</a> . . . . .  | 66 |
| A.2. Diagrama de bifurcación de la actividad del mapa de Rulkov en el plano $(\mu, \alpha)$ . Figura reproducida de (Rulkov, 2002) . . . . .   | 67 |
| C.1. Latencias sin kernel de tiempo real. . . . .  | 75 |
| C.2. Latencias con kernel de tiempo real. . . . .  | 76 |

# Índice de Tablas

|   |    |
|---|----|
| 3.1. Característica de los ordenadores utilizados en el proyecto. . . . .   | 17 |
| 3.2. Característica de las DAQ utilizadas en el proyecto. . . . .   | 18 |
| 3.3. Versiones del software utilizado en el proyecto. . . . .   | 19 |
| 5.1. Número de valores para generar una ráfaga en cada modelo. . . . .  | 34 |
| 5.2. Rango de valores del potencial para los diferentes modelos neuronales y un ejemplo<br>de neurona viva. . . . . | 35 |
| 6.1. Resumen de los pasos a seguir para conseguir una interacción híbrida. . . . .                                  | 39 |
| A.1. Valores iniciales para el modelo Hindmarsh-Rose . . . . .  | 65 |
| A.2. Parámetros seleccionados inicialmente para el modelo Hindmarsh-Rose . . . . .                                  | 65 |
| A.3. Valores iniciales para el modelo Izhikevich . . . . .  | 65 |
| A.4. Parámetros seleccionados para el modelo Izhikevich . . . . .   | 65 |
| A.5. Valores iniciales para el mapa de Rulkov . . . . .   | 67 |
| A.6. Parámetros seleccionados inicialmente para el mapa de Rulkov . . . . .   | 67 |



# Índice de Códigos

|  |    |
|--|----|
| 6.1. Algoritmo para la detección de eventos en una señal neuronal . . . . .    | 41 |
| 6.2. Algoritmo de calibración temporal. . . . .                                | 42 |
| 6.3. Algoritmo de calibración del potencial de membrana. . . . .               | 43 |
| B.1. Comando para la instalación de Linux PREEMPT_RT . . . . .                 | 70 |
| B.2. Comandos para la compilación de librerías Comedi . . . . .                | 71 |
| B.3. Comandos para descargar e instalar Clamp-cli . . . . .                    | 71 |
| B.4. Guía de argumentos de llamada de Clamp-cli . . . . .                      | 71 |
| B.5. Guía de códigos de la opción de autocalibración -c de Clamp-cli . . . . . | 71 |





# Glosario

- **Burst:** Ráfaga de disparos neuronales
- **Circuito híbrido:** Circuito formado por neuronas vivas biológicas y neuronas artificiales
- **Comedi:** Control and Measurement Device Interface
- **CPG:** Generador Central de Patrones (Central Pattern Generator)
- **DAQ:** Tarjeta de Adquisición de Datos (Data Acquisition Card)
- **DBS:** Estimulación cerebral profunda (Deep brain stimulation)
- **Despolarización:** Aumento del voltaje de membrana
- **ECM:** Error cuadrático medio
- **EDO:** Ecuación diferencial ordinaria
- **Hiperpolarización:** Disminución del voltaje de membrana
- **RTAI:** Real Time Application Interface
- **Sinapsis:** Proceso por el cual las neuronas intercambian información entre ellas
- **Spike:** Disparo neuronal



# 1

## Introducción

### 1.1. Motivación

---

El estudio del sistema nervioso llevado a cabo desde principios del siglo XX (Kandel et al., 2000) ha permitido establecer un conocimiento mínimo sobre diferentes aspectos del sistema nervioso relacionados con su fisiología o funcionalidad. Sin embargo, aún queda mucho que comprender sobre las distintas dinámicas neuronales y sus metodologías para el procesamiento de información (Rabinovich et al., 2006). Para ello, es necesario el desarrollo en paralelo de tecnologías que permitan la correcta interacción con estos sistemas, pudiendo ser así estudiados de manera óptima.

Una metodología clásica basada exclusivamente en la observación de diferentes parámetros como pueden ser el potencial de membrana neuronal, la concentración de calcio o el consumo de oxígeno presenta problemas para caracterizar completamente comportamientos o reproducir situaciones deseadas (Chamorro et al., 2012). Para solucionar estos problemas es posible establecer interacciones artificiales con los sistemas biológicos con el fin de producir y caracterizar los comportamientos o situaciones más relevantes desde la perspectiva de su procesamiento de información. Sin embargo, cabe destacar que los sistemas neuronales trabajan en regímenes temporales concretos que deben ser respetados a la hora de establecer conexiones con ellos.

Por lo anterior, es útil conseguir establecer, mediante técnicas de ciclo cerrado, interacciones híbridas (Broccard et al., 2017; Szücs et al., 2000) que consigan cumplir las restricciones temporales y fisiológicas impuestas por los sistemas vivos, permitiendo, por ejemplo, realizar estimulaciones dependientes de la actividad observada. Sin embargo, estas interacciones híbridas no son sencillas y requieren de diversos requisitos así como de la utilización de conocimientos y técnicas de diferentes campos, haciendo necesaria la multidisciplinaridad (Chamorro et al., 2012).

En este contexto es necesaria la utilización de modelos neuronales que cuenten con un equilibrio entre realismo, para permitir las interacciones híbridas, y simplicidad, pues deben poder ser calculados por debajo de tiempos determinados. Además, se requiere del ajuste dinámico de distintos parámetros que definen el comportamiento de la neurona simulada, de su rango de funcionamiento, de la sinapsis utilizada o que regulan la temporalidad del experimento, para conseguir interacciones híbridas correctas.

Es decir, que a los conocimientos relativos al área de la neurociencia es necesario añadir conocimientos de ingeniería informática y computación que permitan adaptar de manera adecuada las simulaciones realizadas a la realidad del sistema biológico (Muñiz et al., 2009).

En concreto, en este trabajo se han desarrollado técnicas y algoritmos que sirvan para establecer de forma semiautomatizada conexiones híbridas entre modelos neuronales y neuronas vivas de sistemas biológicos. Por un lado, para realizar estas conexiones correctamente debe contemplarse en primer lugar igualar las escalas temporales de los modelos a las biológicas, que cuentan con una granularidad de décimas de milisegundos. Por otro lado los parámetros que regulan el comportamiento del modelo deben ser ajustados de manera adecuada.

Para los ajustes de escalas temporales es necesaria la utilización de tecnologías de tiempo real que permitan respetar las frecuencias de funcionamiento de las neuronas. En concreto, en muchos casos es necesario utilizar un sistema operativo que permita ejecutar programas en tiempo real. También es necesario adaptar estos programas a las necesidades concretas de este paradigma.

Respecto a la adaptación del comportamiento de los modelos, es necesario realizar análisis online de las señales recibidas del sistema biológico, estableciendo de manera dinámica los ajustes a realizar para conseguir que los objetivos de la conexión sean cumplidos o para realizar una determinada acción en un momento concreto de tiempo.

Estos dos requisitos, adaptación temporal y de comportamiento, no pueden ser autoexcluyentes entre ellos, por lo cual es necesario también que los análisis online sean lo más precisos posibles a la par que el tiempo necesario para calcularlos no puede exceder la frecuencia de interacción con el sistema biológico.

Por todo lo anterior, este trabajo trata de establecer una metodología clara para la realización de experimentos con circuitos híbridos, definiendo de manera concreta los pasos necesarios para establecer estas conexiones correctamente. También se utilizará esta metodología con ejemplos concretos de modelos neuronales y de sinapsis, proponiendo algoritmos concretos para calibrar y gestionar automáticamente la interacción según el objetivo del experimento.

En concreto las validaciones con sistemas vivos de este proyecto serán realizadas con circuitos generadores de patrones del ganglio estomatogástrico del cangrejo de mar. Estos circuitos son los responsables de la producción de ritmos motores y entre sus características se encuentran su gran robustez y flexibilidad dinámica (Selverston et al., 2000), por lo que su estudio es interesante para comprender la función de cada elementos del sistema, y en concreto cómo son negociados los ritmos o cómo se transmite la información (Szücs et al., 2000).

---

## 1.2. Objetivos

---

Este Trabajo Fin de Máster tiene como objetivo definir protocolos automatizados para el establecimiento, la calibración y el uso de conexiones híbridas entre modelos computacionales que simulan el potencial de membrana de una neurona y neuronas biológicas procedentes de sistemas vivos. Para ello deben ajustarse diversos parámetros de los modelos neuronales y de sinapsis por un lado y de la temporalidad por otro.

Por lo tanto pueden indicarse como objetivos:

- Definir un protocolo de interacción en ciclo cerrado que permita realizar conexiones híbridas. Además este protocolo debe ser fácilmente extensible, tratando de abstraer los diferentes elementos del mismo lo máximo posible.

- Diseñar y validar algoritmos concretos sobre el procedimiento diseñado, ilustrando como su ejecución permite realizar circuitos híbridos de manera correcta, obteniendo como resultado unos modelos neuronales que, por ejemplo, imitan mejor el comportamiento de un elemento del circuito original. En concreto para las validaciones se realizarán experimentos con el circuito generador de patrones del ganglio estomatogástrico de cangrejos de mar.
- Utilizar las conexiones híbridas como herramienta para el control y la exploración dinámica de experimentos, ya que es posible variar los parámetros del circuito híbrido, para caracterizar o controlar el sistema biológico.

Para cumplir estos objetivos es necesaria la utilización de tecnología de tiempo real, por este motivo deberá de utilizarse un sistema operativo que proporcione esta característica. También deberán utilizarse librerías para la comunicación con tarjetas de adquisición de datos (DAQ).

Los requisitos temporales imponen también la necesidad de realizar una programación eficiente que permita cumplir los tiempos estrictos, lo que impone la utilización del lenguaje de programación C, debido a su alta eficiencia, contar con interfaces de programación para el acceso a la tarjeta de adquisición de datos y ser el utilizado generalmente para la programación en tiempo real.

### 1.3. Estructura de la memoria

---

Esta memoria está organizada en los siguientes capítulos:

- **Estado del arte:** En primer lugar se introduce el campo sobre el que se ha desarrollado el TFM, explicando las técnicas utilizadas para el registro y modelización neuronal. Se introducen también los conceptos necesarios para comprender el resto del trabajo como son los distintos tipos de sinapsis, los circuitos generadores centrales de patrones (CPGs), o los circuitos híbridos. Por último se incluye información sobre procesos en ciclo cerrado y búsqueda dinámica de parámetros.
- **Hardware y software utilizado:** Información relativa a los equipos informáticos utilizados así como su software. También se incluye información sobre las neuronas vivas utilizadas.
- **Diseño:** En este capítulo se explica el protocolo desarrollado para el establecimiento de conexiones híbridas.
- **Desarrollo:** Recoge de manera detallada las diferentes fases del protocolo indicando las soluciones implementadas.
- **Experimentos realizados y resultados:** Resultados de los experimentos realizados utilizando el protocolo desarrollado.
- **Conclusiones y trabajo futuro:** Análisis final de los resultados obtenidos así como futuras vías de investigación.
- **Anexos:** Detalles de los modelos neuronales utilizados, del programa desarrollado, de las latencias del sistema y difusión del trabajo.



# 2

## Estado del arte

### 2.1. El sistema nervioso

---

El sistema nervioso es el encargado de captar, transportar y procesar la información en los animales. Gracias a esto, controla y coordina los demás sistemas y órganos para así conseguir un correcto funcionamiento de todas las funciones del individuo (Netter, 1991).

Este sistema se encuentra fundamentalmente formado por dos tipos de células: las neuronas y las células gliales. Las neuronas son consideradas como la unidad básica a nivel estructural y funcional del sistema nervioso, mientras que las células gliales sirven como soporte a las neuronas, controlando el microambiente neuronal, regulando el nivel de iones y de neurotransmisores, así como el suministro de proteínas (Gómez Nicola and Nieto Sampedro, 2008). A su vez suele ser dividido en tres subsistemas de acuerdo a la función concreta que realiza. Estos tres subsistemas son el sistema sensorial, el sistema motor y el sistema nervioso central.

#### 2.1.1. La neurona

La neurona, elemento con el cual se establece el objetivo de realizar una conexión en este TFM, es una célula con un alto grado de especialización, cuya función es la recepción y transmisión de impulsos nerviosos a las otras neuronas a las cuales se encuentre conectada. La neurona es una de las células con una mayor variabilidad morfológica. Debido a esto, pueden ser clasificadas de acuerdo a su morfología, aunque también existen clasificaciones basadas otros criterios como por ejemplo las funciones que desempeñan o su comportamiento.

La neurona se encuentra formada por un núcleo celular llamado soma, el cual contiene los elementos que se encuentran en todas las células. Por otro lado, la neurona cuenta con ramificaciones llamadas dendritas que sirven como receptores de señales neuronales provenientes de otras neuronas o receptores neuronales. Por último, cuenta con un axón, que es el elemento encargado de la transmisión de las señales producidas a otras neuronas, que a su vez recibirán esta información típicamente por sus correspondientes dendritas (Kandel et al., 2000).

La neurona cuenta con una membrana citoplasmática, que separa su interior del medio en el cual se encuentra. Por medio de esta membrana la neurona tiene una permeabilidad selectiva,

permitiendo la entrada y salida solo de determinadas moléculas iónicas. Según su comportamiento se distinguen canales pasivos y activos. Lo primeros se encuentran permanentemente abiertos mientras que los segundos son regulados por medio de señales eléctricas, químicas o mecánicas, pudiendo así cerrarse o abrirse.

Debido a la membrana entre el interior y el exterior de la neurona se produce una diferencia de potencial eléctrico, conocida como potencial de membrana. El valor de esta diferencia a lo largo del tiempo es utilizado como rasgo más característico de la actividad neuronal. En este proyecto es el valor fisiológico contemplado y es simulado en las neuronas artificiales utilizadas para establecer una conexión híbrida. Cuando se toma como referencia exterior un valor de 0 mV puede establecerse que aproximadamente en reposo el interior de una neurona es de -65 mV (Hille, 1977; Kandel et al., 2000).

Cuando el potencial de membrana supera un umbral de despolarización se produce un potencial de acción, comúnmente conocido como spike. Por último es importante conocer que cuando una neurona alterna episodios de spikes y de reposo se denomina actividad spiking. Además se producen agrupaciones de spikes en periodos de tiempo muy reducidos, a esta actividad se la conoce como actividad en ráfagas (bursting) (Kandel et al., 2000).

### **2.1.2. Sinapsis neuronal**

Como se ha mencionado, las neuronas establecen conexiones por medio de axones y dendritas, que sirven para transmitir los potenciales de acción generados gracias a la liberación de neurotransmisores. Las sinapsis sirven por lo tanto para establecer una unión desde el punto de vista funcional (Hormuzdi et al., 2004).

Las sinapsis pueden ser de diversos tipos. La sinapsis eléctricas son aquellas que se producen entre neuronas que se encuentran a poca distancia, siendo la corriente transmitida de una membrana a otra directamente. Por lo tanto no hay liberación de neurotransmisores. Esta sinapsis es una conexión rápida y no hay retardos en la comunicación. En estas conexiones cada neurona presenta una resistencia a la corriente introducida por la otra neurona, lo que modula la influencia que tiene sobre la neurona postsináptica.

Por el contrario, las sinapsis químicas se producen entre neuronas más alejadas y por medio de la liberación de neurotransmisores. La sinapsis química cuenta con diferencias respecto a la eléctrica tales como su duración, su menor velocidad o que pueden ser graduales. También, mientras que la sinapsis eléctrica implica una sinapsis en sentido contrario esto no tiene porque ser así en una sinapsis química, que puede ser unidireccional. Existen sinapsis químicas cuyos neurotransmisores liberados provocan potenciales de acción en la neurona postsináptica pero también las hay que liberan neurotransmisores que provocan una inhibición en la neurona postsináptica (López-Muñoz et al., 2006). Estas sinapsis, dependiendo de sus características son modeladas de manera diferente. En este TFM en la sección 5.1.2 se indica el funcionamiento de dos modelos y en el anexo A se detallan sus ecuaciones.

## **2.2. Utilidades del estudio de la actividad neuronal**

---

El análisis y comprensión del sistema nervioso y de cómo las neuronas integran, transmiten y procesan información tiene múltiples aplicaciones. Tradicionalmente ha sido estudiado por su interés científico y médico, sin embargo el paso del tiempo ha demostrado que su estudio tiene múltiples aplicaciones.

Atendiendo a la medicina, la comprensión del funcionamiento del sistema nervioso permite el tratamiento de, por ejemplo, enfermedades neurodegenerativas, para lo cual es fundamental



además su conjunción con otros campos debido a las necesidades que impone la experimentación e interacción con los sistemas biológicos.

Por ejemplo, el conocimiento del sistema nervioso es útil para averiguar las causas de una patología, como en el caso de la enfermedad del Parkinson, para la cual se ha determinado que una falta en la producción de dopamina, un neurotransmisor, provoca un desbalance entre inhibiciones y excitaciones, lo que tiene como resultado un incremento de las inhibiciones en un área del tálamo, el GPi, que comienzan a oscilar, produciendo los síntomas de esta enfermedad (temblores, rigidez, lentitud en los movimientos e inestabilidad). Puesto que el problema es causa de una deficiencia de dopamina es posible tratar la enfermedad aportando al paciente dopamina de manera artificial para compensar la deficiencia en su producción. Aunque gracias a conocer también la causa más directa que produce la enfermedad es también posible realizar estimulación cerebral profunda (DBS) sobre las zonas afectadas con el objetivo de sincronizar las poblaciones neuronales, consiguiendo de esta manera mitigar la patología (Schiff, 2010).

En el área de la ingeniería, también es posible encontrar múltiples ejemplos de la utilidad que tiene el estudio de la actividad neuronal. En el caso concreto de los circuitos que son objeto de estudio en este TFM, los circuitos generadores centrales de patrones, su estudio en diversos invertebrados ha permitido simular los modelos de generación de ritmos que producen los osciladores neuronales acoplados con los que cuenta para ser utilizados en robots modulares bioinspirados, con el objetivo de regular su locomoción a la par que se aprovechan las ventajas que ofrecen estos circuitos, como son su resistencia dinámica a fallos o la robustez y negociación del ritmo que producen (Herrero-Carrón et al., 2011).

Incluso la aplicación de los conocimientos adquiridos supera a los campos a primera vista más evidentes como son la medicina o la robótica. Un ejemplo de esto es la utilización de paradigmas basados en el procesamiento de la información neuronal aplicados a campos como la estadística o el aprendizaje automático. Por ejemplo las redes neuronales artificiales (Glorot and Bengio, 2010) son usadas a menudo en problemas de diversa índole como pueden ser de optimización o clasificación, utilizando principios biológicos tales como la regla de Hebb (Hebb, 1949). Un ejemplo de problema concreto para los cuales las redes neuronales artificiales resultan muy adecuadas son los traductores de idiomas (Bahdanau et al., 2014).

---

## **2.3. Observación de la actividad neuronal**

---

Para registrar la actividad neuronal y establecer interacciones con sistemas biológicos se han desarrollado a lo largo de los años una serie de técnicas destinadas a, principalmente, medir los potenciales neuronales y a inyectar corriente, neurotransmisores o neuromoduladores.

Para realizar las mediciones existe la posibilidad de realizarlas de dos maneras:

- **Extracelular:**

La información registrada corresponde a la señal agregada de todas las señales que sean captadas en la zona de medición. Por ejemplo, los potenciales de todos axones que incluye un nervio que está registrando el electrodo.

- **Intracelular:**

Introduciendo un electrodo en el interior de la neurona es posible realizar mediciones de la variación del potencial de membrana entre el medio y el interior de la neurona.

Las técnicas desarrolladas pueden, en general, clasificarse en varios tipos atendiendo a la forma de registro y control de señalización neuronal:

- *Voltage clamp* (pinzamiento de voltaje):

Gracias a fijar el valor de voltaje se consigue medir la corriente eléctrica, es decir, el paso de iones por la membrana neuronal, permitiendo caracterizar este valor (Roth et al., 2014; Destexhe and Bal, 2009). Este es el método de medida que fue desarrollado en primer lugar por Kenneth Cole y George Marmont durante sus experimentos con el axón gigante del calamar (Cole and Marmont, 1942).

- *Current clamp* (pinzamiento de corriente):

Este método es similar al anterior, sin embargo, en esta técnica el valor fijado es la corriente, lo que permite medir el voltaje.

- *Dynamic clamp* (pinzamiento dinámico):

Con el objetivo de poder desarrollar interacciones híbridas es posible inyectar corriente en la célula en función de su potencial de membrana. Es por lo tanto la técnica más completa, ya que realiza una interacción que permite, por ejemplo, producir sinapsis artificiales consiguiendo así inducir comportamientos determinados o construir circuitos híbridos (Prinz et al., 2004; Broccard et al., 2017).

Dada la intención de realizar circuitos híbridos con modelos neuronales en este proyecto es necesario la utilización de registros intracelulares. En este caso se requiere inyectar corriente a la par que se registra el potencial de la membrana para implementar las conexiones que llegan tanto a las neuronas vivas como a los modelos neuronales, por lo cual la técnica utilizada es la de pinzamiento dinámico, contando con un electrodo en el medio extracelular como referencia y dos en el interior de la célula. Uno de los electrodos intracelulares sirve para registrar el potencial mientras que el otro será utilizado para la inyección de corriente.

## 2.4. Simulación de la actividad neuronal

---

Desde que se consiguieron los primeros registros de actividad neuronal se han realizado a la par modelos con los cuales conseguir reproducir y caracterizar los resultados obtenidos.

Cole y Curtis realizaron experimentos en los cuales pudieron registrar el potencial de membrana del axón gigante del calamar (Cole and Curtis, 1939). Sumado esto a los conocimientos de cuáles eran los canales con mayores concentraciones de iones (potasio, sodio y cloro) y por lo tanto cuáles son los canales que caracterizan en mayor proporción el potencial les permitió realizar un circuito electrónico cuya señal de salida reprodujera la actividad observada. En este circuito (figura 2.1) las resistencias equivalen a los canales iónicos, las fuentes de tensión a los gradientes de tensión de los iones y el condensador a la capacidad de carga eléctrica de la membrana.

Adicionalmente, durante estos primeros años también se desarrollaron modelos menos realistas fisiológicamente pero más centrados en el procesamiento de la información. El es caso de los trabajos de McCulloch y Pitts (McCulloch and Pitts, 1943), Frank Rosenblatt (Rosenblatt, 1958) o Donald Hebb (Hebb, 1949).

Retomando el circuito de Cole y Curtis cabe destacar que estos circuitos pueden naturalmente representarse por medio de ecuaciones. Hodgkin y Huxley, que también trabajaban con el axón gigante del calamar al igual que Cole y Curtis, realizaron un modelo neuronal por medio de ecuaciones diferenciales que permite reproducir el potencial de membrana neuronal e inyectar corriente en la neurona (Hodgkin and Huxley, 1952). Este primer modelo tenía una representación directa con la realidad, constando de una ecuación para el potencial de membrana y una ecuación por cada una de las conductancias iónicas tenidas en cuenta.

En los años siguientes surgieron un número importante de variaciones de este modelo, añadiendo diferentes canales o ajustes para representar diferentes comportamientos o neuronas determinadas (Connor et al., 1977). Algunos de estos modelos que cabe destacar como ejemplos son el modelo de FitzHugh-Nagumo (Nagumo et al., 1962) o de Kepler (Kepler et al., 1992), que buscan ser una versión simplificada del original o el modelo de Morris-Lecar (Morris and Lecar, 1981) que está desarrollado para imitar la fibra muscular gigante del percebe y ha sido utilizado para describir el comportamiento de neuronas del ganglio estomatogástrico de la langosta (Skinner et al., 1993) o neuronas sensoriales de mamíferos (Morris and Lecar, 1981).

En general los modelos de tipo Hodgkin-Huxley resultan pesados computacionalmente ofreciendo un alto grado de realismo. Por este motivo su uso en circuitos híbridos puede necesitar técnicas como la tabulación que permitan acelerar el tiempo de cálculo durante la ejecución. Sin embargo, es posible utilizar modelos más sencillos computacionalmente pero que conserven un grado de realismo suficiente.

Buscando un modelo que conservara el realismo a la par que redujera el coste computacional y aumentara la versatilidad Hindmarsh y Rose presentaron un modelo (Hindmarsh and Rose, 1984) que utiliza tres ecuaciones diferenciales no lineales con un coste computacional menor a la vez que es capaz de reflejar actividad spiking o bursting.

A partir de este modelo la tendencia ha continuado y existen modelos que han avanzado aún más en el objetivo de reducir la complejidad computacional conservando un potencial de acción realista, aunque a costa de perder la descripción de los canales iónicos. Esto, a priori, presenta ventajas debido a la posibilidad de realizar mejores simulaciones de, por ejemplo, grandes poblaciones neuronales si el objetivo es atender al procesamiento de la información en la red sin atender a los detalles de la biofísica subcelular. Si, en cambio, se busca el estudio de los canales sinápticos estos modelos modernos pueden no ser adecuados.

El modelo que más destaca respecto a esta idea es el modelo de Izhikevich (Izhikevich, 2003) que reduce enormemente el coste computacional y cuenta solo con dos ecuaciones diferenciales. También existen modelos que hacen uso de otras ideas, como el modelo de Rulkov (Rulkov, 2002) que utiliza un mapa iterado de dos dimensiones, contando con una señal que representa la dinámica lenta y otra que representa la dinámica rápida.

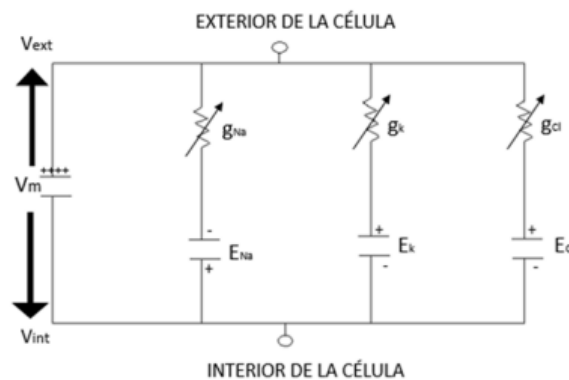


Figura 2.1: Circuito que simula el comportamiento pasivo de una membrana neuronal con tres canales.

## **2.5. Interacción en ciclo cerrado**

---

Una metodología de ciclo cerrado hace referencia a un proceso de feedback en el cual de acuerdo a la salida del sistema observado introduce un estímulo, es decir, que se aplica una entrada en el sistema de acuerdo al comportamiento del propio sistema tras un procesamiento de la información de la señal observada. De esta manera se consigue una estimulación dependiente de la actividad que se está monitorizando.

En lo referente a este proyecto, el proceso de ciclo cerrado será realizado sobre una neurona de un sistema biológico. En este caso la señal introducida de nuevo a la neurona viva será calculada mediante modelos artificiales de neuronas y sinapsis. De esta manera, es posible la caracterización del comportamiento neuronal, permitiendo estudiar las propiedades de la dinámica neuronal durante esta interacción.

Las neuronas y por ende las redes nerviosas forman un sistema que es no lineal, realizando un procesamiento de la información en régimen transitorio, por lo cual la interacción en ciclo cerrado y dependiente de los estímulos es un método natural para estudiar estos sistemas.

La interacción en ciclo cerrado requiere utilizar diversas tecnologías y técnicas. Es necesaria la utilización de la técnica de pinzamiento dinámico ya que se requiere la lectura de voltaje y la inyección de corriente. Además, estas lecturas y envíos de información deben de realizarse con periodicidad estricta, por lo cual es necesario contar con tecnología de tiempo real que permita cumplir estos requisitos (Chamorro et al., 2009; Muñiz et al., 2009; Ortega et al., 2014).

Esta metodología ya ha sido utilizada con el objetivo de manejar los circuitos centrales generadores de patrones (Sharp et al., 1992; Elson et al., 1998; Varona et al., 2001a). Estos circuitos, formados por neuronas, generan ritmos regulares, sin embargo, cuando una de las neuronas es aislada del resto del circuito esta regularidad se rompe. Gracias a una interacción en ciclo cerrado es posible por medio de la lectura del potencial de membrana de la neurona inyectar corriente de manera precisa, consiguiendo de esta manera regularizar el ritmo de disparo de la neurona aislada.

## **2.6. Circuitos híbridos**

---

Como se ha descrito en los capítulos anteriores, es posible realizar lecturas instantáneas del potencial de neurona, así como inyectar corriente en la neurona por medio de la técnica de pinzamiento dinámico. Si a esto sumamos los modelos neuronales y la interacción en ciclo cerrado, que plantea un estímulo en el sistema pero dependiente de las lecturas realizadas, es posible la construcción de circuitos híbridos (Broccard et al., 2017).

Las primeras implementaciones de circuitos híbridos se realizaron con neuronas electrónicas, es decir, implementando estos modelos mediante circuitos electrónicos. Las conexiones entre estas neuronas electrónicas y las vivas resulto satisfactoria, demostrando la utilidad de estos circuitos. Por ejemplo, las primeras publicaciones que utilizaron circuitos híbridos sirvieron para determinar la funcionalidad de las oscilaciones subumbrales de las neuronas de la oliva inferior (Yarom, 1991) o para comprender las interacciones entre las neuronas de un CPG (Szücs et al., 2000; Elson et al., 1998; Varona et al., 2001a) observando la importancia de la conexión bidireccional de los elementos del circuito para producir los ritmos regulares.

Otros ejemplos de utilización de circuitos híbridos incluyen estudios del funcionamiento de zonas del cerebro como puede ser el tálamo (Le Masson et al., 2002) permitiendo, por ejemplo, simular una entrada visual desde la retina, sirviendo para determinar que existe un valor crítico en la ganancia del bucle de retroalimentación talámico tras el cual el circuito tiende a oscilar, provocando una decorrelación temporal entre la entrada desde la retina al tálamo y la salida.

En resumen, el uso de circuitos híbridos es una manera efectiva de caracterizar los sistemas neuronales (Broccard et al., 2017), permitiendo analizar regímenes concretos de funcionamiento del sistema dinámico, así como permitir controlar de manera precisa la actividad del sistema de manera dependiente a la actividad observada. Es importante, además, resaltar que los circuitos híbridos han demostrado su eficacia como vía para el estudio de la utilidad y validez de los modelos neurales (Linaro et al., 2015; Oprisan et al., 2004; Nowotny et al., 2003)

### 2.6.1. Esquema general de un circuito híbrido

La construcción de un circuito híbrido requiere la utilización de modelos neuronales, pero estos deben ser adaptados adecuadamente para poder ser utilizados de manera correcta en esta tarea. En primer lugar cabe destacar que es necesario adaptar temporalmente los modelos para que su escala temporal sea similar a la biológico. Adicionalmente, si un modelo tiene una escala en amplitud de señal diferente a la que presenta la neurona viva también será necesario adaptar la señal para que sea adecuada durante las interacciones. Tradicionalmente, esto se hacía estableciendo escalas temporales y amplitudes fijas en la neurona electrónica construida, sin embargo, el uso de neuronas software que plantea este proyecto ofrece una mayor flexibilidad en este proceso que puede ser realizado de acuerdo a la señal observada.

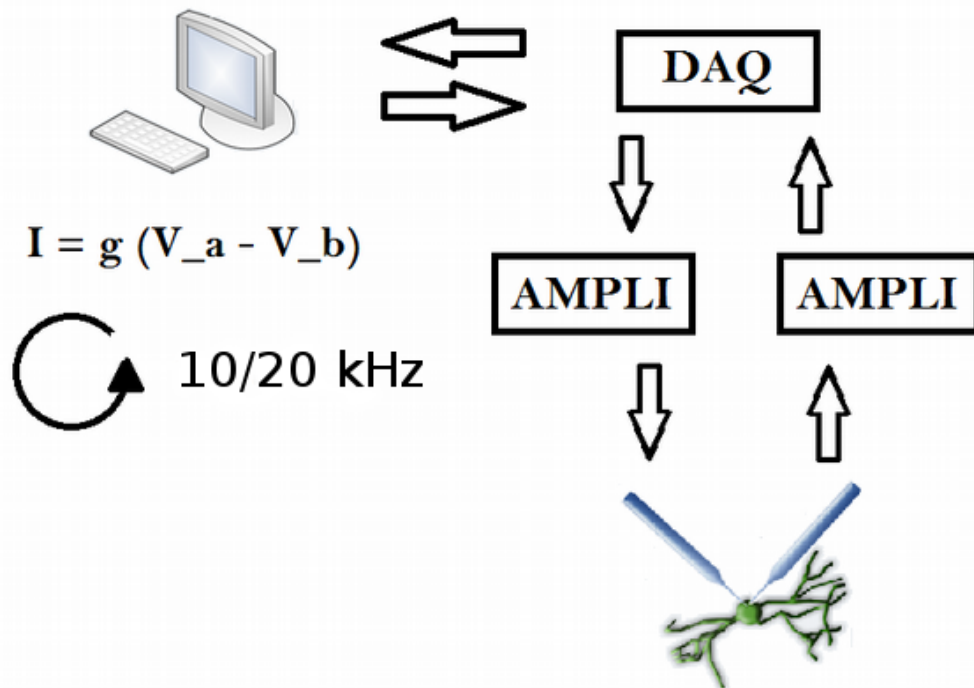


Figura 2.2: Esquema de construcción de una interacción en ciclo cerrado y un circuito híbrido.

Por otro lado, esta conexión híbrida, dado el objetivo de imitar la conexión natural entre neuronas, debe contar con una sinapsis que articule y regule la conexión. Este proceso también debe ser calibrado para conseguir una interacción correcta.

Este proceso se ilustra en la figura 2.2, donde se representa un circuito híbrido en el cual la interacción entre neurona viva y modelo tiene una frecuencia de 10 o 20 kHz. Una vez cada periodo se realiza una lectura del potencial de membrana por medio de la tarjeta de adquisición de datos. Con este valor se inyecta, por medio de una sinapsis artificial, el valor en el modelo

neuronal que ejecuta el ordenador, que realiza los cálculos necesarios (detallados en los capítulos sucesivos de esta memoria) para producir el voltaje de membrana de la neurona artificial. Una vez obtenido este valor, se realiza el proceso inverso y este potencial se convierte en una corriente a través del modelo de sinapsis, siendo este valor enviado a la tarjeta que a su vez por medio del amplificador inyecta la corriente en la neurona viva.

El proceso descrito presenta diversas problemáticas, tales como adaptar las escalas temporales o de valores de los modelos utilizados o ajustar el funcionamiento de los modelos neuronales para conseguir comportamientos concretos. Además se requiere de conocimientos de programación y la utilización de tecnología en tiempo real. Debido a esto, a pesar de ofrecer grandes posibilidades, los circuitos híbridos no han sido a día de hoy utilizados de una manera generalizada. En este contexto, este trabajo pretende conseguir un desarrollo y una definición clara de los procesos necesarios, incluyendo algoritmos de automatización, para conseguir adaptar modelos neuronales para su uso en circuitos híbridos.

---

## 2.7. Circuitos Generadores Centrales de Patrones

---

Los Circuitos Generadores Centrales de Patrones (abreviados como CPG debido a su nombre en inglés, *Central Pattern Generator*), son los circuitos encargados de la coordinación de los músculos involucrados en un movimiento constante, como puede ser la locomoción, respirar, masticar o los movimientos digestivos (Harris-Warrick, 2011). Las motoneuronas de estos circuitos dirigen los movimientos de los músculos consiguiendo una gran flexibilidad para adaptar el ritmo producido a las circunstancias cambiantes que puedan producirse.

Los circuitos híbridos sobre los que se habla principalmente en la sección anterior y en los cuales han sido probados los protocolos del presente TFM son CPGs, debido a su accesibilidad por un lado y por el interés en el estudio de la regularidad y flexibilidad de los ritmos producidos por otro lado.

Debido a la relativa simplicidad de estos circuitos, así como la sencillez de su acceso fisiológico respecto a otros circuitos neuronales los CPG son un candidato idóneo para la realización de diferentes experimentos, tales como la construcción de circuitos híbridos, el estudio de regímenes y comportamientos neuronales o el efecto de distintos neurotransmisores.

Ejemplos de esto son los hitos conseguidos con estos circuitos como el acoplamiento de dos neuronas vivas de manera artificial (Elson et al., 1998) o la construcción de circuitos híbridos (Szücs et al., 2000; Pinto et al., 2000). Estos trabajos permitieron comprobar que las neuronas aisladas del sistema producen ritmos de ráfagas de distinta duración entre ellos de manera aleatoria y que es la conexión de los elementos del sistema lo que conduce a la producción de un ritmo regular.

Estos trabajos pueden servir para desarrollar modelos concretos sobre un tipo de neuronas (Elices and Varona, 2015; Varona et al., 2001b), permitiendo a su vez un estudio mayor debido a la posibilidad de realizar, por ejemplo, simulaciones computacionales. También es posible descubrir y analizar de manera concreta propiedades intrínsecas de la red como puede ser la dependencia que existe entre ritmo y fase (Hooper et al., 2015).

Finalmente, los CPGs también pueden ser utilizados para estudiar la viabilidad de los circuitos híbridos como herramienta para la reparación de problemas en el comportamiento o la sustitución de elementos dañados en circuitos que generan ritmos robustos y flexibles. En esta línea, el primer circuito híbrido en un CPG permitió restaurar los ritmos regulares que previamente habían sido dañados (Szücs et al., 2000).

## 2.8. Control de la actividad y exploración dinámica automática

---

Como se ha mencionado, el proceso de realización de circuitos híbridos no es trivial y requiere de diversas adaptaciones durante el establecimiento de la conexión. Por lo tanto, es necesaria la realización de protocolos que establezcan de manera dinámica en cada experimento los valores de los modelos neuronales y sinápticos así como de la conexión que conduzcan a maximizar el objetivo planteado de acuerdo a un criterio.

La utilidad de esta calibración puede ser analizada desde diversas perspectivas. En primer lugar, resulta interesante analizar por qué existe la necesidad de realizar esta calibración en sí misma. Como se ha mencionado en las secciones anteriores, la construcción de circuitos híbridos es interesante por diversos motivos entre los que se incluye comprender las mecánicas neuronales o controlar la actividad de una red (ver secciones 2.5 y 2.6).

Por otro lado, desde un punto de vista práctico, que la calibración sea automática y se produzca en el mínimo tiempo posible presenta ventajas respecto a una búsqueda manual de los parámetros de conexión. Por ejemplo, reduce los riesgos para la preparación biológica, puesto que es posible implementar mecanismos de protección, estableciendo límites sobre los rangos de funcionamiento. Además, estos experimentos son difíciles de realizar debido a la dificultad de la disección y debe aprovecharse al máximo los que salgan correctamente. A esto se suma el tiempo limitado de vida del elemento a estudiar una vez extraído del resto del organismo, por lo que no perder tiempo en la búsqueda de parámetros resulta muy ventajoso.

Tradicionalmente, esta calibración y adaptación de modelos neuronales para su uso en circuitos híbridos ha seguido un enfoque hardware (sección 2.6) lo cual presenta limitaciones a la hora de modular el ciclo cerrado de la conexión. En contraposición la realización de neuronas software como se plantea en este TFM permite una mayor flexibilidad de las neuronas y sinapsis implementadas, que pueden alterar su funcionamiento con una mayor facilidad. Además, este enfoque software permite el análisis online de los resultados, permitiendo en la propia interacción ajustar la conexión para obtener un resultado concreto buscado, la exploración dinámica y la automatización de los experimentos.

La calibración automática software requiere, además, de la utilización de tecnologías diversas como es el análisis de resultados en ciclo cerrado y la utilización de tecnología de tiempo real, a la par que conocimientos sobre modelos y dinámicas neuronales.

### 2.8.1. Ejemplos de control y calibración automática

Los ritmos exactos o el envío de información en tiempos precisos son requeridos dado que ciertas neuronas solo responden a estructuras temporales concretas, siendo este un factor importante en la dinámica neuronal (Tiesinga et al., 2008). Por esto si se desea, por ejemplo, establecer un ritmo concreto es necesario realizar una calibración temporal del modelo y también de las constantes de tiempo de las sinapsis. La automatización de este proceso puede servir por tanto para maximizar la respuesta del sistema consiguiendo, por ejemplo, recuperar en preparaciones de laboratorio comportamientos dañados durante la preparación del experimento o sirviendo como mecanismo de sustitución de elementos dañados de un sistema.

Por ejemplo, es posible mediante un algoritmo que analiza en tiempo real la señal biológica obtenida controlar el tiempo de spike, consiguiendo de esta manera sincronizar neuronas oscilatorias de la región CA1 del hipocampo (Stigen et al., 2011). De esta manera es posible además, establecer una fase concreta entre la neurona biológica y la inyección de corriente, sirviendo por ejemplo para establecer disparos en antifase o una sincronización del momento de disparo. En este ejemplo concreto el control se establece mediante la relación funcional que existe entre la

amplitud de los pulsos aplicados a una fase concreta y la fase avanzada. Es posible utilizar esta relación como medida del objetivo a conseguir observando, por ejemplo, cómo responde esta medida al cambio de otros valores tales como la amplitud del estímulo.

Si vamos más allá del control y tratamos los circuitos híbridos, la calibración también es muy útil para establecer de manera dinámica los parámetros que mejor determinan el comportamiento de modelo neuronal para un objetivo, como puede ser la sincronización, frecuencia de disparo o regularidad del ritmo.

Esta calibración requiere realizar detección de eventos sobre la señal, como puede ser la detección de spikes o ráfagas, para conseguir una determinada frecuencia o valores iguales entre un modelo y una neurona viva. Para esto, es necesario la utilización de técnicas de tratamiento de señal así como análisis concretos para señales electrofisiológicas (Mulansky et al., 2015).

Para la calibración de los modelos neuronas debe tenerse en cuenta su condición de sistemas no lineales. En estos sistemas la búsqueda de parámetros no es fácil y requiere la realización de barridos sobre los rangos posibles de un valor observando el efecto producido en el sistema tras cada cambio. Para esto existen técnicas de búsqueda y estimación en sistemas no lineales (Liao et al., 2017) que pueden ser consideradas siempre que no entren en conflicto con los requisitos estrictos de temporalidad que presenta la interacción con neuronas biológicas.

Para establecer cuál es el mejor valor es posible determinar una métrica que indique la similitud entre dos señales, pudiendo ser una de ellas la biológica y otra la artificial. Esta métrica puede ser una medida estadística tal como el error cuadrático medio u otras a explorar. Una vez determinada la similitud puede iterativamente explorarse un nuevo valor (Nowotny and Varona, 2012). Es decir, mediante la comparación de la métrica obtenida entre la configuración antigua y la nueva se elige la más óptima, repitiéndose el proceso de nuevo hasta conseguir un comportamiento del modelo neuronal que encaje con el observado en la neurona biológica.

### **2.8.2. Accesibilidad de la calibración automática**

Como se ha mencionado, los procesos y análisis a realizar en tiempo real sobre registros neuronales no resultan fáciles y necesitan del esfuerzo compartido de diferentes campos de conocimiento. Además, en último término el objetivo es la realización de herramientas útiles para experimentos de electrofisiología, que son realizados por personal científico que no cuenta necesariamente con los conocimientos en informática o procesamiento de señal que son requeridos. Por estos motivos es necesario conseguir incorporar las técnicas desarrolladas a plataformas que faciliten su utilización.

Es decir, que dada la utilidad que tienen los circuitos híbridos y su calibración automática en el campo de la neurociencia resulta fundamental que estos puedan ser utilizados de manera intuitiva por personal ajeno al desarrollo del protocolo interno. Además, estas herramientas también pueden servir para la automatización de experimentos, permitiendo definir de manera concreta periodos de estimulación, cambios en el sistema, etc.

Existen ejemplos de plataformas que cuentan con interfaz gráfica o con una utilización sencilla mediante comandos para la realización de experimentos que utilicen técnicas de dynamic clamp (Kemenes et al., 2011). Estos programas engloban a la vez el manejo del resto de herramientas necesarias revisadas en este estado del arte, como es la programación en tiempo real y el manejo de las tarjetas de adquisición de datos (Muñiz et al., 2009; Linaro et al., 2014; Biró and Giugliano, 2015). Sin embargo estas herramientas no cuentan con protocolos de calibración y exploración automática de los parámetros que regulan los circuitos híbridos.



## 2.9. Sistemas operativos de tiempo real

---

Para la correcta interacción con los sistemas biológicos se requiere la lectura y envío de información con una periodicidad estricta. La frecuencia requerida en estos casos es de alrededor de 10.000 o 20.000 Hz. Un ordenador actual puede soportar estas frecuencias con soltura, realizando además los cálculos necesarios entre cada periodo, sin embargo el software y los sistemas operativos de propósito general no están pensados para cumplir estas restricciones temporales. Esto es debido a diversos motivos, como las interrupciones de procesos del sistema operativo que tienen mayor prioridad, el tener que compartir recursos tales como el disco duro, o la preferencia a mostrar eventos como el movimiento del ratón en tiempo real para conseguir una mejor experiencia del usuario medio.

Estos problemas pueden ser solucionados mediante la utilización de un sistema operativo en tiempo real que permita que nuestro programa sea ejecutado con prioridad y tenga acceso preferente a los recursos requeridos. Existen diversas alternativas a la hora de seleccionar un sistema operativo en tiempo real, tanto libres como privativas con un elevado coste, sin embargo, se han considerado solo soluciones que trabajan sobre el kernel Linux y son de uso libre. Esto es debido por un lado a que las herramientas de programación de sistemas GNU/Linux son las más accesibles y potentes. Por otro lado, la utilización de software libre reduce costes para el grupo de investigación a la vez que permite una facilidad mayor de la difusión del trabajo y su utilización por otros laboratorios.

Sistemas de tiempo real sobre el kernel Linux considerados:

- RTAI: permite al kernel de Linux soportar tiempo real (Beal et al., 2000). Es el proyecto más antiguo de los contemplados y usado hasta la fecha en el laboratorio del Grupo de Neurocomputación Biológica (GNB). Sin embargo su desarrollo es lento (tardando en aparecer las versiones compatibles con los kernels más reciente de Linux). Además, requiere realizar una compilación difícil y su utilización requiere la programación en espacio de kernel para funcionar de la mejor manera posible.
- Xenomai: este proyecto ofrece una modificación del kernel de Linux (Gerum, 2004) que sigue la estela de RTAI respecto a su estructura, contando de manera general a la hora de programar tanto con sus ventajas como desventajas.
- Real-Time Linux (PREEMPT\_RT): la implementación oficial de tiempo real en Linux. Tradicionalmente ha ofrecido resultados peores a las otras soluciones presentadas, sin embargo en los últimos años ha mejorado hasta poder ser considerada una alternativa a tener en cuenta (Oliveira and Oliveira, 2015). Cuenta con ventajas como que no necesita la utilización de una librería concreta para la realización de programas en tiempo real, ya que son utilizadas las propias de cualquier sistema Linux y una instalación mucho más fácil, así como soporte para versiones recientes del kernel.

El objetivo de este trabajo no consiste en seleccionar cuál es el adecuado pero sí que es deseable que el seleccionado para el futuro desarrollo sea el que ofrezca el mejor balance entre correcto funcionamiento y una facilidad de instalación y programación razonable. Se ha escogido por cumplir estas características PREEMPT\_RT, ya que es el más fácil de instalar y utilizar, tiene mayor soporte y proyección de futuro y cumple a priori con las latencias que son requeridas para la realización de circuitos híbridos.



# 3

## Hardware, software y preparación biológica

Dado el carácter experimental del trabajo es necesario indicar de manera precisa los elementos utilizados para la realización de los experimentos. Esto permite que los resultados puedan ser replicados teniendo en cuenta el hardware y software utilizado.

En el caso del hardware es importante debido a las limitaciones de frecuencia o rango de funcionamiento que marca el fabricante de las tarjetas de adquisición de datos. Respecto al software, es necesario conocer la versión de las diferentes soluciones utilizadas, pues su calidad podría variar en un futuro. Por último, también es muy importante indicar sobre que neuronas se realizan los experimentos y a qué especie corresponden.

### 3.1. Hardware

---

Se ha comprobado que los algoritmos pueden correr en el tiempo requerido en distintos ordenadores y se ha determinado que, si bien las latencias pueden variar, la totalidad de ordenadores modernos así como ordenadores con diez de antigüedad (Intel Pentium 4) pueden cumplir los requisitos para ejecutar correctamente el protocolo diseñado.

#### 3.1.1. Ordenadores

En la tabla 3.1 se recogen las características de los equipos utilizados. La puntuación benchmark se ha obtenido de [www.cpubenchmark.net](http://www.cpubenchmark.net).

|                           | Ordenador 1             | Ordenador 2                    | Ordenador 3               |
|---------------------------|-------------------------|--------------------------------|---------------------------|
| Procesador                | Intel Pentium 4 3.2 GHz | Intel Core 2 Quad Q9650 3.0GHz | Intel Core i7-6700 3.4GHz |
| Benchmark                 | 376                     | 4.226                          | 10.038                    |
| Núcleos (físicos/lógicos) | 1/1                     | 1/1                            | 4/8                       |
| Almacenamiento            | HDD                     | HDD                            | SSD                       |

Cuadro 3.1: Característica de los ordenadores utilizados en el proyecto.

### 3.1.2. Tarjetas de adquisición de datos

Cada ordenador se encuentra conectado a una tarjeta de adquisición de datos (DAQ). El GNB cuenta con tarjetas de la marca National Instruments (tabla 3.2) y para su manejo en GNU/Linux se utilizan las librerías Comedi ([www.comedi.org](http://www.comedi.org)). La numeración de la tarjeta en la tabla corresponde con la del ordenador a la cual se encuentran conectadas en el resto de tablas de este capítulo.

Estas tarjetas se conectan en este caso al ordenador mediante PCI, aunque otros modelos de tarjetas pueden utilizar otra conexión como USB. A su vez estas tarjetas están conectadas a un bornero que cuenta con puertos BNC que sirven como entrada o salida.

En este proyecto los programas realizados han realizado lecturas y escrituras de la tarjeta de datos con una frecuencia de 10.000 o 20.000 Hz.

|       | Modelo                                     | Rango de voltajes (voltios)         |
|-------|--|-------------------------------------|
| DAQ 1 | National Instruments PCI-6529              | 10, 5, 2, 1, 0.5, 0.2, 0.1          |
| DAQ 2 | National Instruments PCI-MIO-16E-4 (6040E) | 10, 5, 2.5, 1, 0.5, 0.25, 0.1, 0.05 |
| DAQ 3 | National Instruments PCI PCI-6251          | 10, 5, 2, 1, 0.5, 0.2, 0.1          |

Cuadro 3.2: Característica de las DAQ utilizadas en el proyecto.

### 3.1.3. Neuronas electrónicas

Durante el desarrollo del software y como test para los experimentos se han realizado pruebas con neuronas electrónicas. El GNB cuenta con dos (ver sección 2.4) que implementan el modelo Hindmarsh-Rose, con una temporalidad equiparable a la que presentan las neuronas biológicas. Además permiten generar comportamientos de actividad en ráfagas irregulares, útil para los experimentos de negociación de ritmos.

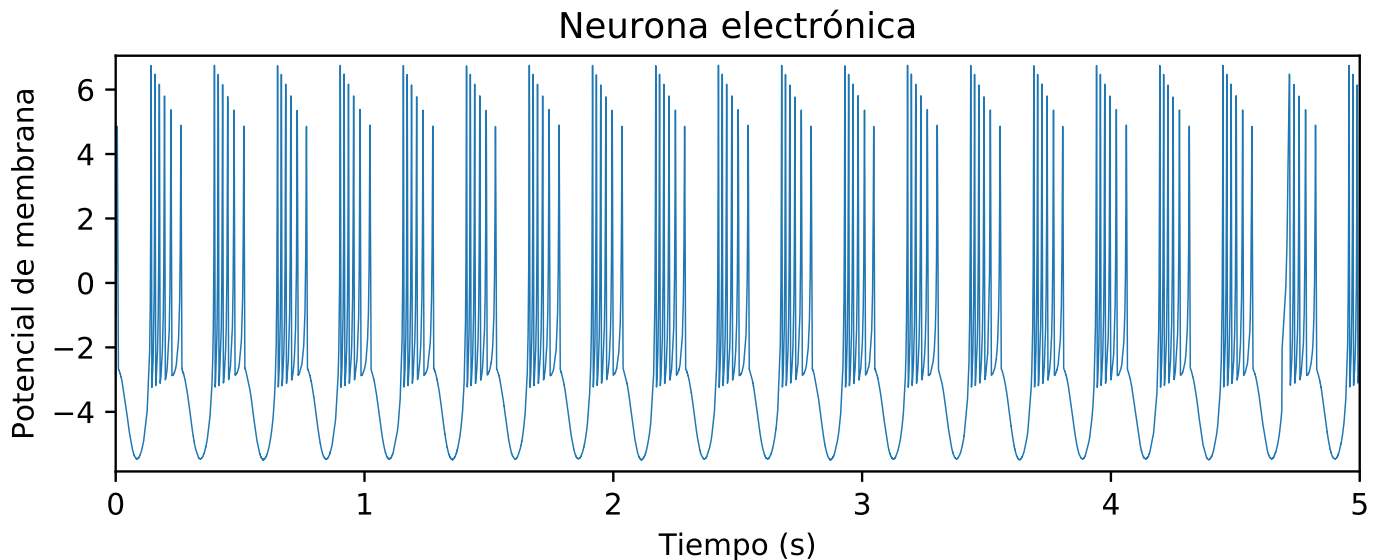


Figura 3.1: Señal producida por una neurona electrónica del GNB.

### 3.2. Software

Se detalla en la tabla 3.3 las versiones del sistema, de las librerías y soluciones de tiempo real utilizadas en cada ordenador.

|                   | Ordenador 1   | Ordenador 2 | Ordenador 3      |
|-------------------|---------------|-------------|------------------|
| Sistema operativo | Ubuntu 10.04  | Fedora 7    | Debian 9         |
| Kernel Linux      | 2.6.35.7-rtai | 2.6.20-rtai | 4.9.0-2-rt-amd64 |
| Tiempo real       | RTAI          | RTAI        | PREEMPT_RT       |
| Driver Comedi     | ni_pcmio      | ni_pcmio    | ni_pcmio         |
| Versión Comedi    | 0.7.76        | 0.7.74      | 0.10.2-4         |

Cuadro 3.3: Versiones del software utilizado en el proyecto.

### 3.3. Neuronas vivas

Las neuronas con las cuales se han realizado circuitos híbridos son extraídas del ganglio es-tomatogástrico de cangrejos de mar (*Carcinus maenas*), en concreto se trata de motoneuronas encargadas de la coordinación de los movimientos intestinales, es decir, forman parte de un circuito generador central de patrones (ver sección 2.7). Las neuronas de este circuito son grandes y resistentes a la inyección de electrodos. La preparación puede durar varias horas e incluso mantenerse viva durante varios días.

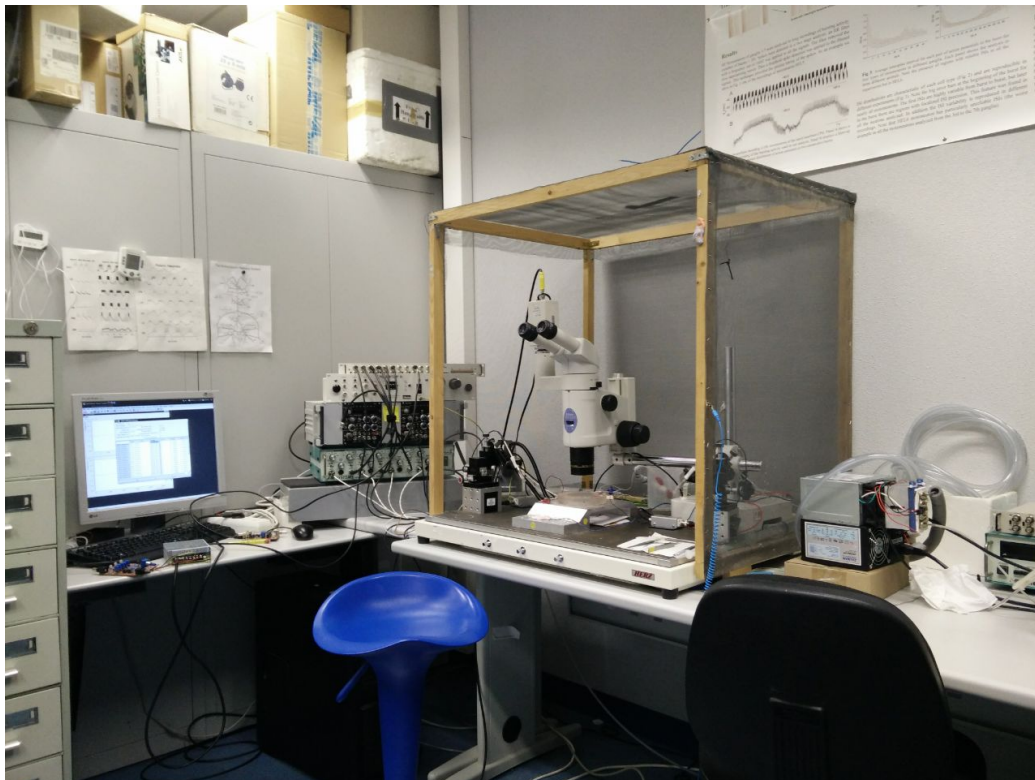


Figura 3.2: Fotografía del setup de electrofisiología del Grupo de Neurocomputación Biológica (GNB).

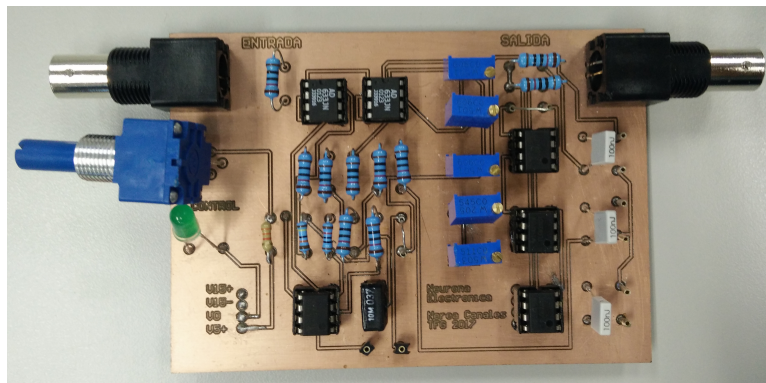


Figura 3.3: Fotografía de una neurona electrónica.



Figura 3.4: Cangrejo de mar común (*Carcinus maenas*) y detalle de la primera parte de su disección (estómago abierto) para los experimentos realizados.

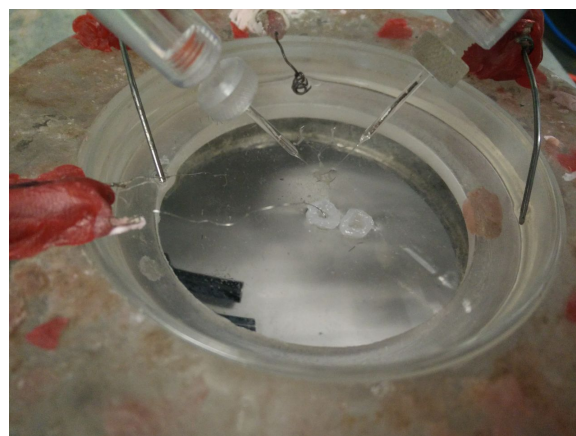
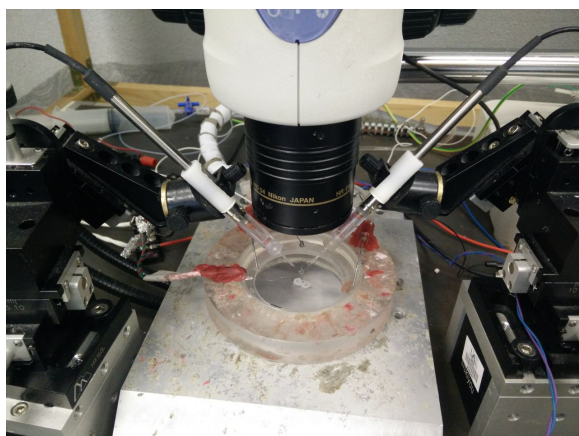


Figura 3.5: Ganglio estomatogástrico con electrodos conectados a una neurona para registrar la actividad neuronal e inyectar corriente.



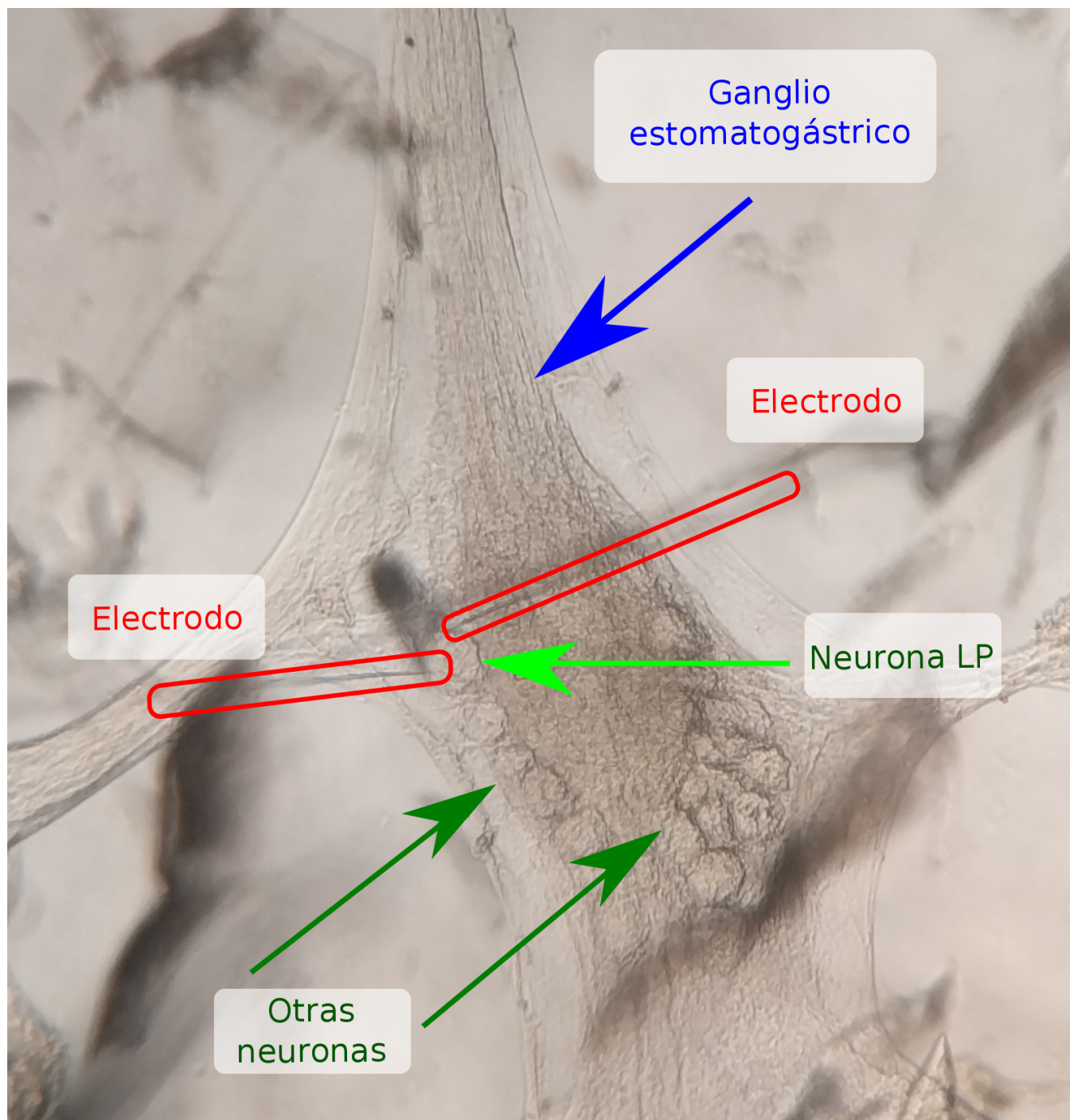


Figura 3.6: Imagen del microscopio tras un experimento. Se indican los elementos observados.





# 4

## Diseño

La realización de circuito híbridos requiere de la conjunción de distintos elementos y tecnologías que deben ser utilizadas de manera conjunta correctamente para conseguir unos resultados correctos. Por este motivo en este Trabajo Fin de Máster se ha diseñado en primer lugar un esquema general de los distintos elementos que forman parte del procedimiento necesario.

### 4.1. Planteamiento inicial

---

Como primera aproximación puede visualizarse el ordenador que procesa la información leída y decide la corriente a inyectar como una caja negra como se ilustra en la figura 2.2. Como puede observarse, el ordenador, a través de una tarjeta de adquisición de datos, se encuentra conectada por medio de unos amplificadores y electrodos a una neurona para establecer una conexión por medio de la técnica de dynamic clamp (ver sección 2.3).

Por las características de las interacciones neuronales, se hace necesario que el ordenador en primer lugar reciba y envíe información con una frecuencia estricta que debe ser respetada, como se observa en la figura 4.1.

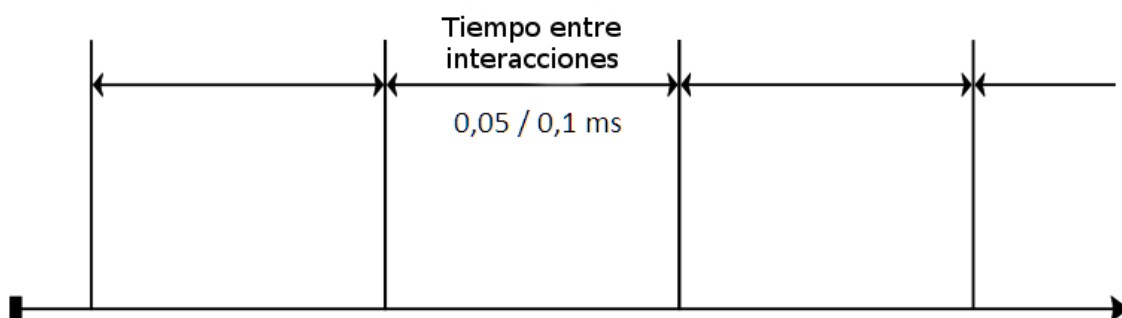


Figura 4.1: Esquema que representa el periodo constante de interacción con el sistema biológico y el tiempo disponible para realizar cálculos entre cada interacción a frecuencias de adquisición/envío de 10-20 kHz.

Es decir, que el tiempo entre cada una de las interacciones es el tiempo máximo del que dispone el ordenador para realizar todos los cálculos que sean necesarios para en función de la lectura realizada al comienzo producir una salida de acuerdo a los criterios fijados.

A partir de esto podemos determinar los dos primeros factores que delimitan el procedimiento y los cálculos posteriores. En primer lugar, como se ha mencionado, los cálculos realizados no pueden sobrepasar en ningún caso el tiempo entre interacciones. En segundo lugar, si este tiempo de cálculo es menor que el máximo, el envío del resultado y la nueva lectura de la tarjeta no deben producirse hasta el momento requerido, por lo cual es necesario un método de espera funcional y exacto para la granularidad de tiempo contemplada (0.05 o 0.1 milisegundos como tiempo entre interacciones). Esta frecuencia está determinada por las frecuencias soportadas por las tarjetas de adquisición de datos del GNB (ver sección 3.1.2).

## 4.2. Esquema propuesto

Una vez determinadas las condiciones externas en las cuales se enmarcan las interacciones del ordenador con el exterior el siguiente paso es establecer un esquema de los procesos internos que deben llevarse a cabo para conseguir conexiones neuronales híbridas entre los modelos neuronales que ejecutará el ordenador y las neuronas vivas de manera correcta.

Para ello se han establecido a grandes rasgos dos subsistemas. Un primer subsistema se encarga de realizar análisis previos de la señal que está siendo registrada de la neurona viva, así como de los modelos y sinapsis a utilizar. Cabe destacar que el análisis de los modelos no cambia entre ejecuciones por lo cual puede ser calculada de en cada ejecución (como sucede con la señal biológica) o puede estar guardada de un análisis previo. Durante esta fase es posible determinar además si el sistema es capaz de cumplir correctamente con las latencias requeridas o si la tarjeta de adquisición de datos se encuentra correctamente conectada.

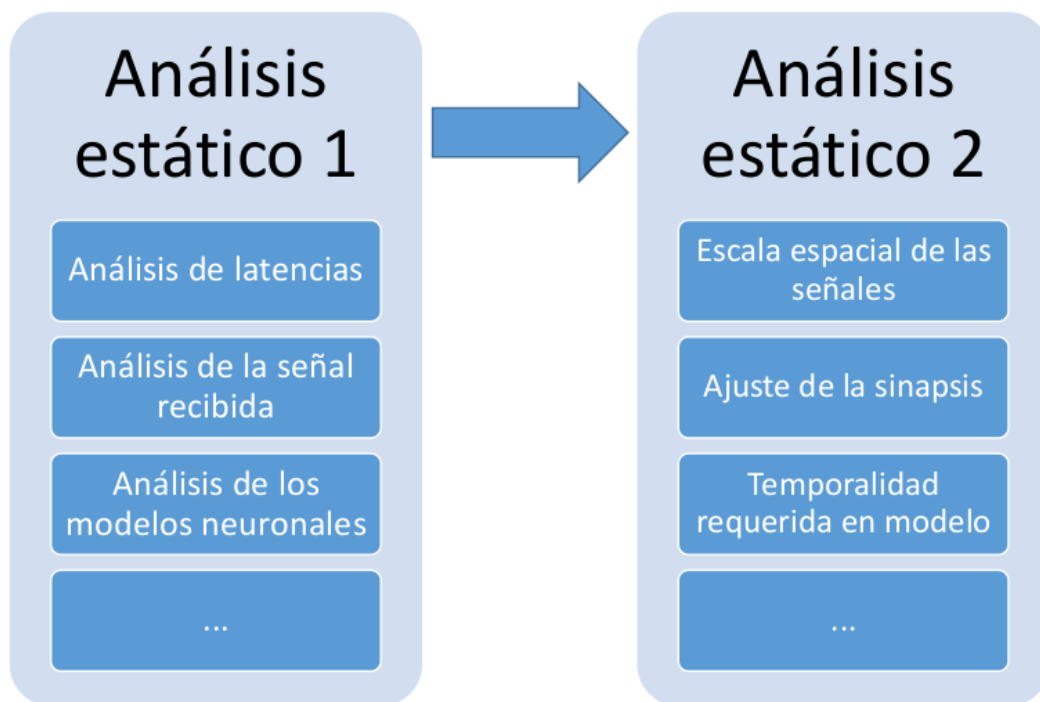


Figura 4.2: Primer subsistema del diseño planteado.

Esta primera fase puede ser planteada como se indica en la figura 4.2 estableciendo un primer periodo de observación para, a continuación, establecer los valores requeridos mediante la información observada.

En concreto, los análisis de latencias sirven para ofrecer un feedback al usuario sobre la validez o no de los datos a recoger y avisar de problemas. El análisis de la señal recibida consiste en un periodo de observación en el cual se determinan valores de la señal como puede ser su rango de valores, la velocidad de los disparos neuronales, esto es, su frecuencia de disparo, o el nivel de ruido presente en la señal. Respecto a los modelos, es necesario caracterizar su rango en amplitud o establecer cuántos puntos de simulación es necesario calcular para conseguir una determinada estructura identificable, como por ejemplo, una ráfaga de disparos. Gracias a esta información es posible determinar cómo establecer una conexión correcta, que será refinada durante el segundo subsistema.

La segunda fase es la relativa a la propia interacción híbrida e incluye todos los procesos y cálculos necesarios para que esta se produzca correctamente, como es el cálculo del modelo neuronal, su escalado, el ajuste dinámico de parámetros o el envío y recepción de información.



Figura 4.3: Segundo subsistema del diseño planteado.

En la figura 4.3 se ilustra el esquema seguido, en el cual cada periodo es iniciado mediante la lectura y escritura de información de la tarjeta de adquisición de datos. Tras esto, se realizan todos los procesos que deben ser ejecutados en cada interacción. Los valores determinados durante el primer subsistema son utilizados por los diferentes elementos del segundo subsistema. Por otro lado, durante estos periodos también es posible realizar modificaciones en los parámetros de acuerdo a uno o varios criterios de referencia con el objetivo de que sean maximizados. En este proyecto este objetivo consiste principalmente la correcta sincronización neuronal, para lo cual pueden ser utilizadas medidas estadísticas sobre la semejanza de ambas señales o de relación de fase por ejemplo.

Componentes del segundo subsistema:

- Envío y lectura de información de la tarjeta de adquisición de datos.
- Integrar el modelo software: es necesario calcular un número de puntos determinado en cada interacción para igualar el ritmo de ambas señales.
- Detección de eventos y análisis: referente al cálculo de las mencionadas medidas estadísticas de las señales, referencias de fases, corrientes inyectadas o cualquier métrica que sea contemplada.
- Obtener medida de la sincronización u otro objetivo: integración de la información del punto anterior.
- Cambio de parámetros: realizar cambios en los valores que definen el comportamiento del modelo o la conexión si la medida de sincronización definida aún no ha sido satisfecha, con el objetivo de obtener un resultado mejor en la próxima interacción.
- Determinar la corriente a inyectar en la próxima interacción: calcular por medio de la sinapsis utilizada el valor de corriente que se producirá hacia la neurona viva y hacia la neurona biológica, así como el escalado necesario y la conversión al formato utilizado por la DAQ.
- Espera hasta el momento determinado: los momentos de interacción con el sistema deben ser respetados, por lo cual el tiempo sobrante hasta el siguiente instante fijado debe ser completado igualmente.

Por último cabe destacar que los pasos indicados no tienen porque realizarse en todas las interacciones, siendo cada módulo autónomo en cuanto a su ejecución. Por un lado es evidente que la integración del modelo, o la inyección de corriente y espera serán realizadas en todos los periodos, pero en cambio la medida de sincronización puede ser realizada sobre una ventana definida por la observación de una ráfaga o un periodo específico de tiempo, consistiendo el resultado en la integración de la información recogida durante todo el periodo.

---

### 4.3. Requisitos

---

La experimentación con neuronas biológicas requiere cumplir ciertas restricciones para conseguir una interacción correcta y que en el proceso el modelo neuronal incluido en el sistema vivo actúe como una neurona biológica. Si no se respetan estos requisitos los resultados obtenidos no serán válidos, pues el objetivo es conseguir registrar interacciones y comportamientos realistas de los sistemas neuronales observados.

A continuación se enumeran las principales restricciones:

- Frecuencia de interacción: la frecuencia de envío y lectura de información debe respetarse escrupulosamente, debiendo situarse entre los 10.000 y 20.000 Hz.
- Tiempo de cálculo: con el objetivo de respetar la frecuencia de interacción el tiempo de cálculo entre los periodos no puede superar en ningún caso la duración de un periodo.
- Realismo del modelo neuronal: Aunque no es necesario que las ecuaciones del modelo definan estrictamente el comportamiento de los canales iónicos de una neurona si es necesario que el potencial de membrana producido sea realista y que el modelo permita la inyección de una corriente externa con una respuesta adecuada a dicho estímulo.
- Respetar rangos de funcionamiento: la interacción debe producirse en una escala de valores adecuada para el sistema vivo que respete el funcionamiento natural de los canales iónicos y que no corresponda a un régimen de seguimiento pasivo de las corrientes inyectadas.



# 5

## Desarrollo

### 5.1. Simulación de la dinámica neuronal

---

Con el objetivo de reproducir la actividad neuronal observada de manera experimental diversos modelos han sido implementados (véase sección 2.4 y anexo A). A continuación se incluyen descripciones de los modelos utilizados en el proyecto de manera general. Los detalles concretos como sus ecuaciones, rango de los parámetros según comportamientos o métodos de integración se indican en el anexo A.

#### 5.1.1. Modelos neuronales

Los modelos neuronales contemplados en este proyecto deben producir un potencial de membrana realista para conseguir una correcta conexión híbrida. Por otro lado, no es necesario generalmente que sean modelos realistas respecto a la simulación de los canales iónicos. Esto es positivo puesto que la simulación realista de estos canales tiene un alto coste computacional (Torres and Varona, 2012), lo cual podría provocar un tiempo de cálculo mayor del disponible (figura 4.1).

Por estos motivos se han seleccionados modelos que a la par que son ligeros ofrecen potenciales de acción realistas y son versátiles. A continuación se describen de manera general los modelos utilizados. Sus ecuaciones, así como parámetros de configuración iniciales determinados se incluyen en el apéndice A. Este apéndice incluye información también sobre como resolver las ecuaciones diferenciales ordinarias que tienen los modelos de Hindmarsh-Rose e Izhikevich.

#### Modelo Hindmarsh-Rose

La primera aproximación a un modelo que ofreciera un buen balance entre rapidez de cálculo y realismo fue realizado por Hindmarsh y Rose (Hindmarsh and Rose, 1984). Esto fue posible gracias a que sustituyeron los canales iónicos que se habían simulado hasta entonces en modelos tipo Hodgkin-Huxley por dos ecuaciones correspondientes a las dinámicas lenta y rápida de la actividad neuronal respectivamente.

Con este modelo es posible reproducir comportamientos de disparos aislados (spike) o ráfagas (burst). Además puede producir un comportamiento caótico, produciendo ráfagas de duración variable. Esto resulta de utilidad puesto que las neuronas del ganglio estomatogástrico del cangrejo utilizadas cuando se encuentran aisladas también presentan un comportamiento irregular.

Cabe destacar también que es un modelo considerado rígido en cuanto a los periodos de duración de la hiperpolarización, lo cual significa que es menos plástico a la hora de negociar ritmos y de adaptar su comportamiento para hacer coincidir de manera exacta sus disparos a los de la neurona a la cual se encuentra conectada. Es el modelo que implementan las neuronas electrónicas del GNB (ver sección 3.1.3).

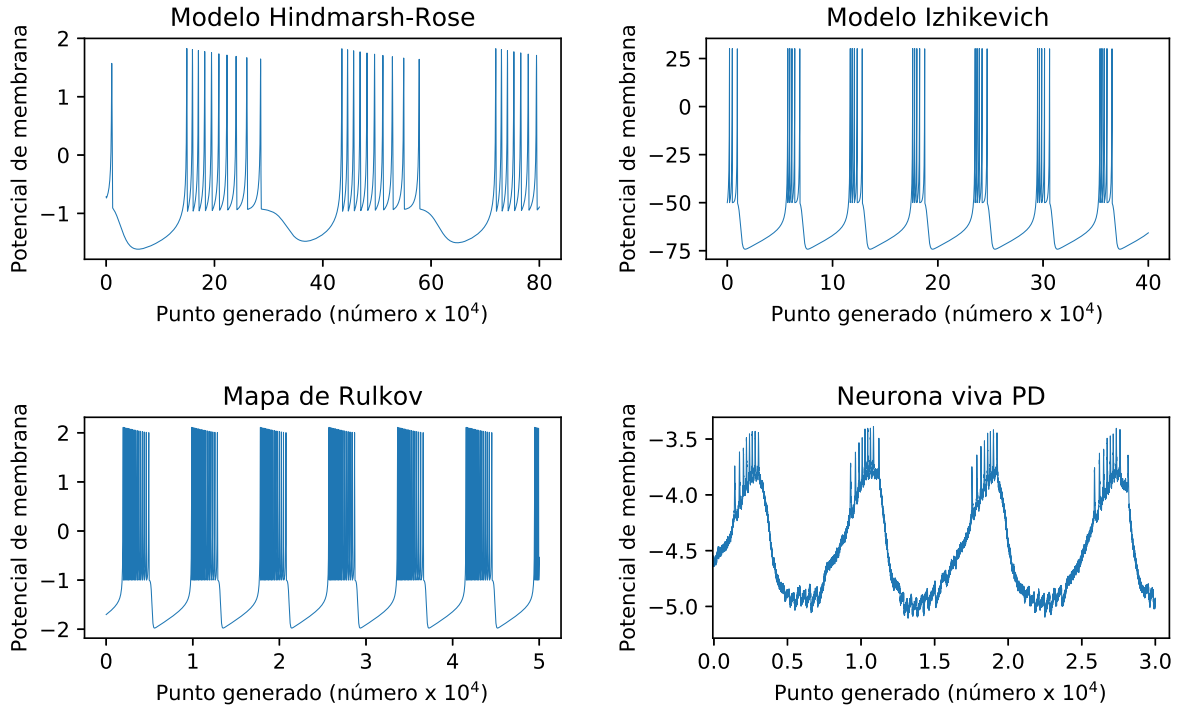


Figura 5.1: Ejemplos de los potenciales de membrana producidos por los modelos neuronales y por una neurona viva del CPG pilórico de un cangrejo.

## Modelo Izhikevich

Durante los años posteriores a la publicación del modelo de Hindmarsh y Rose se profundizó en el análisis de los sistemas no lineales y la actividad neuronal. Se consiguieron así modelos de aún menor carga computacional, que reproducen de mejor manera la plasticidad neuronal y que son capaces de representar dinámicas de neuronas concretas o actuar como resonadores.

Izhikevich plantea un modelo con las citadas características (Izhikevich, 2003), compuesto por dos ecuaciones, una menos respecto al modelo de Hindmarsh y Rose, y una condición de reinicio que depende del potencial de membrana. Ajustando los parámetros del modelo de acuerdo a un mapa aportado por el autor es posible conseguir diferentes comportamientos, algunos añadidos respecto a las posibilidades que ofrece el modelo de Hindmarsh y Rose, como pueden ser las oscilaciones subumbrales.



## Mapa de Rulkov

Siguiendo con la simplificación planteada Rulkov plantea un modelo que no requiere del cálculo de ecuaciones diferenciales ordinarias (EDO), sirviéndose de un mapa iterado. Además este modelo produce menos puntos para representar una ráfaga que el mínimo necesario, por lo cual debe realizarse una interpolación con el fin de aumentar la resolución de la señal producida y que se produzcan suficientes puntos del modelo.

En el modo de utilización de ráfagas este modelo permite variar cuantos potenciales componen una ráfaga o cuál es la frecuencia de estas.

### 5.1.2. Modelos de sinapsis

Las conexiones entre neuronas reciben el nombre de sinapsis. Es decir que los modelos neuronales expuestos en la sección anterior deben conectarse mediante algún tipo de sinapsis con las neuronas biológicas. Las sinapsis se modelan mediante descripciones de corrientes que se inyectan a las neuronas vivas. En los modelos se introducen como una corriente en el lado derecho de las ecuaciones que describen el potencial de membrana (ver anexo A).

#### Sinapsis eléctrica

Este modelo de sinapsis es el más simple y se produce por el paso de iones entre las dos células que componen la conexión, sin necesidad de neurotransmisores (Hormuzdi et al., 2004). Estas reciben el nombre de uniones gap y se producen en células que se encuentran a una distancia muy cercana (alrededor de 3.5 nanómetros).

Para reproducir de manera artificial esta sinapsis es necesario calcular en ambos sentidos de la conexión la cantidad de corriente que una neurona inyecta en la otra. Esto se define por la diferencia de ambos potenciales de la neurona y multiplicado por la conductancia del acoplamiento (la inversa de la resistencia eléctrica). Este proceso se ilustra en la figura 5.2.

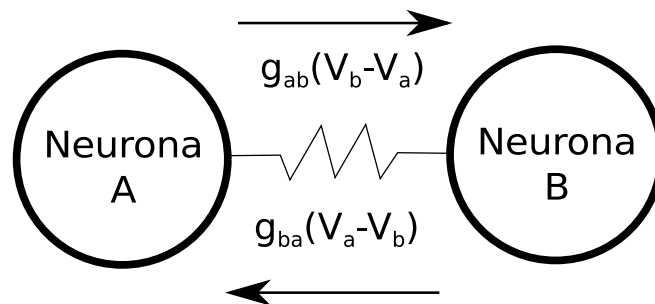


Figura 5.2: Esquema de una sinapsis eléctrica.

#### Sinapsis química

Cuando las neuronas se encuentran a mayor distancia (20-40 nanómetros) el intercambio de información entre las neuronas debe producirse mediante neurotransmisores químicos. Debido a esto, una sinapsis de este tipo puede producirse de manera unidireccional. Por también es un proceso más lento y que requiere de un mayor tiempo entre disparos, pues la neurona que dispara debe restaurar sus niveles de neurotransmisores antes de poder disparar de nuevo.

Las sinapsis químicas pueden tener distintas temporalidades de acuerdo al ritmo de esta liberación de neurotransmisores, así como el ritmo de recuperación. Para ellos existen diversas maneras de reproducir este proceso.

La sinapsis química puede ser simulada por medio de distintos modelos. Por ejemplo, existen modelos de sinapsis gradual (Golowasch et al., 1999) en el cual la corriente de la sinapsis depende del potencial de membrana de las neuronas presináptica y postsináptica.

Otro ejemplo es la sinapsis dinámica, utilizada para caracterizar una conexión sináptica de acuerdo a su cantidad de recursos (Tsodyks and Markram, 1997; Destexhe et al., 1994). Estos recursos son los neurotransmisores que se liberan tras cada sinapsis y que requieren tiempo para ser recargados. Para esto se definen tres estados: efectivo, inactivo y recuperado. Cada conexión presináptica activa una fracción de los recursos disponibles que se encuentran en el estado de recuperado. Tras unos milisegundos en esta situación pasan a estar inactivos y tras un tiempo de alrededor de un segundo los recursos son recuperados.

Estos modelos de sinapsis tienen una mayor cantidad de elementos a ajustar que la sinapsis eléctrica, por lo que la utilización de búsqueda de parámetros en ciclo cerrado es, en este caso, muy útil.

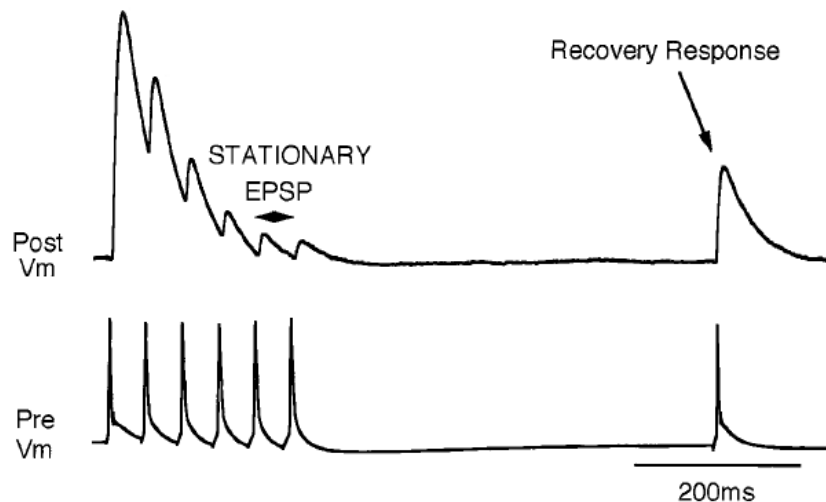


Figura 5.3: Ilustración de una sinapsis dinámica. Figura adaptada de (Tsodyks and Markram, 1997). Se observa la reducción en la efectividad de la sinapsis en la neurona post sináptica.

## 5.2. Primer subsistema

### 5.2.1. Análisis de las señales

#### Frecuencia del disparo

Para establecer la correcta temporalidad de los modelos neuronales respecto a la señal biológica debe conocerse la frecuencia de las ráfagas. Para esto, se han desarrollado sistemas que trantan de lidiar con el ruido en las señales. Los filtros utilizados pueden ser necesarios también en el segundo subsistema, por lo cual se han diseñado con el objetivo de que puedan ser utilizados online, sin embargo, durante esta fase, podría ser utilizado un filtro de la señal offline.

Para filtrar la señal en tiempo real se han desarrollado dos métodos. El primero realiza una convolución de la señal de acuerdo a los valores previos, el segundo, por otro lado, aplica una reducción en el número de valores de la señal, sustituyendo conjuntos consecutivos de valores por la media de estos. Además es posible combinar ambos métodos y realizar un segundo filtrado sobre el primero (o aplicar por ejemplo en dos ocasiones la convolución).

Tras esto para conseguir fijar la frecuencia se realiza una detección de las ráfagas producidas por la neurona. Para esto se han implementado también dos algoritmos. El primero requiere de un valor tras el cual se considera que se ha producido un disparo. Tras producirse un disparo, la detección queda desactivada hasta que se produce un valor menor al del umbral, momento en el cual vuelve a activarse. Como este método puede presentar problemas si la señal es muy ruidosa se ha implementado un método con dos umbrales (de despolarización e hiperpolarización respectivamente). Gracias al doble umbral se consigue una mayor robustez en los resultados ya que no se producen falsas detecciones de eventos (ver capítulo 6).

## Temporalidad

La producción de los valores de un modelo no presenta a priori una dimensión temporal concreta (ver figura 5.4), pues esta depende simplemente de la capacidad de cálculo de nuestro ordenador. Esto marca un requisito mínimo para los modelos candidatos a ser implementados que deben ser capaces de realizar una ráfaga en un tiempo inferior a la décima de milisegundo con frecuencias de muestreo de 10.000 Hz o de media décima de milisegundo con frecuencias de 20.000 Hz.

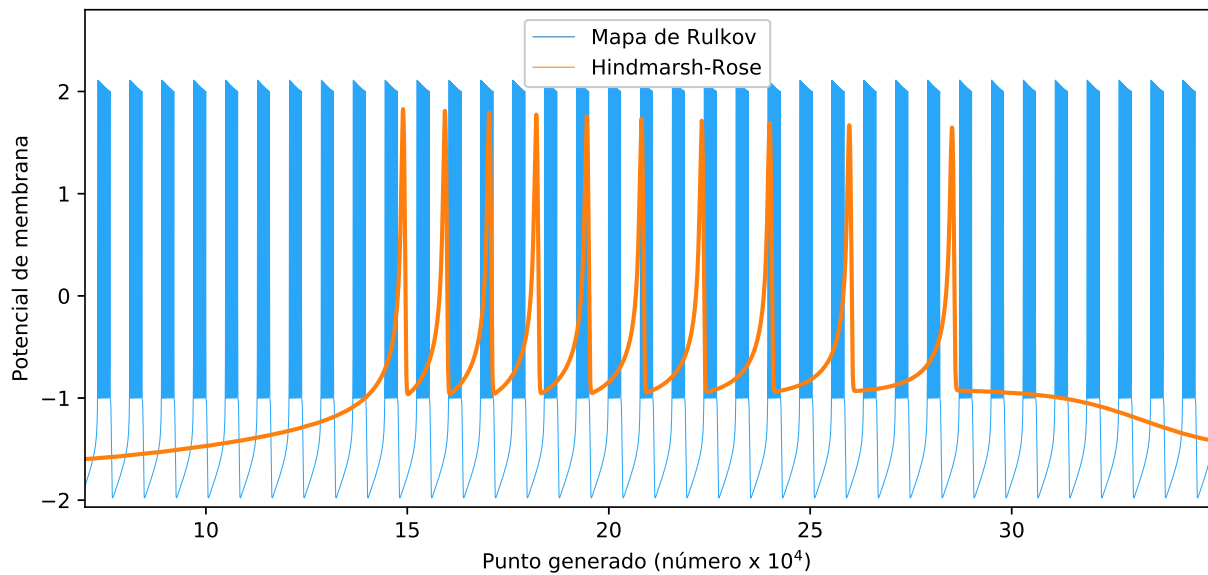


Figura 5.4: Diferencia en el número de puntos para generar una ráfaga entre dos modelos diferentes.

Por otro lado, una vez se ha comprobado que esta limitación se cumple es necesario calcular cuantos puntos requieren cada uno de los modelos para representar una ráfaga de disparos (tabla 5.1). Las neuronas vivan son las que marcan la referencia. Desde una frecuencia de adquisición de datos por medio de la DAQ y gracias a la detección de ráfagas es posible determinar el número de puntos por ráfaga objetivo. Los modelos deberán ser adaptados para conseguir un valor similar. Para esto se presentan dos alternativas:

- Modelo con más puntos de los necesarios: en este caso es necesario descartar puntos. Para ello se calcula la relación de puntos entre el modelo y el objetivo (puntos modelo / puntos objetivo) obteniendo así el número de puntos a calcular entre cada interacción durante el segundo subsistema (utilizando únicamente el último punto para la comunicación).
- Modelo con menos puntos de los necesarios: si por el contrario no se cuenta con suficientes puntos debe optarse por conseguir más puntos a partir de la información disponible. Para ello se ha utilizado una interpolación lineal (ver anexo A).

|                 | Paso de integración | Frecuencia de lectura de la tarjeta (Hz) | Puntos / ráfaga |
|-----------------|---------------------|--|-----------------|
| Hindmarsh-Rose  | 0,001               | -  | 260.166         |
| Izhikevich      | 0,001               | -  | 59.324          |
| Mapa de Rulkov  | -                   | -  | 350             |
| Neurona viva PD | -                   | 10.000                                   | ~7.500          |
| Neurona viva PD | -                   | 20.000                                   | ~15.000         |

Cuadro 5.1: Número de valores para generar una ráfaga en cada modelo.

## Rango de voltaje

Antes de plantearse conectar un modelo neuronal implementado en software con una neurona viva es necesario contemplar que ambas señales deben encontrarse en escalas compatibles. Para ello es necesario realizar un análisis de los valores producidos por cada señal. Como se puede observar en la tabla 5.2, cada una de los modelos contemplados así como un ejemplo de la señal leída de una neurona viva presentan valores completamente diferentes. Un ejemplo de este problema puede ser observado en la figura 5.5 donde se observa que tanto la longitud del voltaje como el rango es diferente.

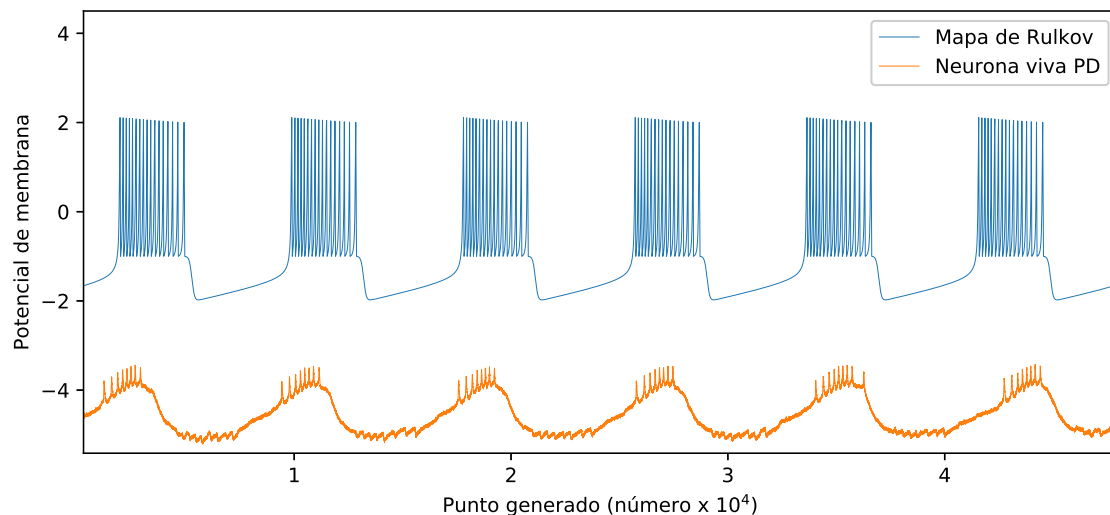


Figura 5.5: Ejemplo de diferencia de escala entre un modelo neuronal y la señal biológica.

Para obtener estos datos al comienzo de la ejecución del protocolo, debe realizarse una observación de la señal biológica para determinar los valores indicados en la tabla 5.2. En el caso de los modelos neuronales puede ser también calculada pero dado que son datos constantes es más conveniente que sean calculados una vez e incluidos como valores de la configuración del modelo.

|                 | Valor máximo | Valor mínimo |
|-----------------|--------------|--------------|
| Hindmarsh-Rose  | 1.797        | -1.609       |
| Izhikevich      | 30.240       | -74.235      |
| Mapa de Rulkov  | 2.111        | -1.977       |
| Neurona viva PD | -3.6         | -5.1         |

Cuadro 5.2: Rango de valores del potencial para los diferentes modelos neuronales y un ejemplo de neurona viva.

### 5.2.2. Cálculo del escalado temporal y de potencial

Con la información adquirida en el punto anterior es posible establecer los parámetros principales que servirán para establecer una parte importante de la conexión híbrida, habilitando que ambas señales sean capaces de interoperar de manera correcta (quedando pendiente para el segundo subsistema regular parámetros de comportamiento del modelo y de la sinapsis).

En definitiva con la información conseguida es posible determinar:

- Puntos relevantes para lograr ráfagas de una duración adecuada de acuerdo al análisis de frecuencia de las ráfagas observadas.
- Escalado del potencial: es necesario adaptar ambas señales a la señal contraria. Es decir, ambas señales seguirán trabajando con sus rangos normales de funcionamiento, sin embargo, la información enviada en cada sentido será escalada. Para esto es necesario determinar valores que ajusten la amplitud de las señales así como su offset relativo.

## 5.3. Segundo subsistema

---

Este subsistema se encarga de realizar el circuito híbrido y de producir una interacción en ciclo cerrado, para ello se sigue el esquema planteado en la figura 4.3. Los componentes de este subsistema requieren de los resultados del primer subsistema para poder funcionar.

### 5.3.1. Integración del modelo neuronal

Durante un periodo de la interacción se calculan los puntos necesario para acoplar la temporalidad del modelo neuronal ejecutado a la de la neurona viva o, en caso de ser insuficientes para reproducir la resolución de la frecuencia de muestreo, se realiza una interpolación lineal (ver anexo A).

Para realizar este proceso el voltaje registrado de la neurona viva es escalado para tener el rango de las unidades del modelo antes de ser utilizado en la sinapsis para obtener la corriente a inyectar en el modelo. Por otro lado, los resultados de voltaje producidos son escalados al rango de la neurona viva para ser utilizados en la sinapsis y que la corriente resultante tenga los rangos adecuados.

### 5.3.2. Análisis de las señales y medida de la sincronización

Para cada par de puntos modelo-vivo se calcula la información relevante y necesaria. Esto puede ser por ejemplo una resta de los valores para observar su diferencia.

Las medidas de la sincronización por su parte se ejecutan tras un periodo de tiempo determinado o tras la sucesión de varios eventos (ráfagas, valores atípicos...) y sirven para integrar la información del análisis progresivo de cada punto de la señal.

En este proyecto se plantea la utilización del error cuadrático medio (ECM), la varianza o la diferencia de fase entre ambas señales, sin embargo cualquier medida que indique la diferencia o similitud de las señales puede ser incluida fácilmente en el protocolo diseñado debido a su modularidad.

Las medidas estadísticas como el ECM o la varianza pueden ser utilizadas en conexiones en las que se busque que ambas neuronas disparen al mismo tiempo (en fase), debido a que se busca incrementar la similitud de ambas. La observación además debe de hacerse sobre un periodo de tiempo que abarque varias ráfagas.

Por otro lado, la diferencia de fase hace uso de los detectores de disparos explicados para el primer subsistema (sección 5.2.1). Mientras que, para las medidas estadísticas, el valor de la sincronización era directamente el resultado obtenido en este caso debe aplicarse de manera directa la varianza debido a que lo se busca es constancia del tiempo entre disparo de ambas neuronas. Cabe mencionar también que esta métrica sirve también para una sinapsis inhibitoria (disparo en antifase).

### **5.3.3. Ajuste dinámico de parámetros**

Las métricas obtenidas en el apartado anterior son utilizadas finalmente para variar de manera dinámica parámetros que definen el comportamiento del modelo neuronal en ejecución o de la sinapsis. Para esto es posible establecer diferentes maneras de tener en cuenta el valor de sincronización obtenido:

- Umbral: alcanzado un determinado valor prefijado se considera que la conexión conseguida es correcta. Este objetivo puede ser fijado como un porcentaje respecto a una primera situación de no sincronización.
- Pendiente: se analiza el nivel de disminución o aumento de la métrica hasta que no se produce o la mejora es mínima.
- Diferencia entre registros: en vez de utilizar la pendiente del valor puede calcularse la varianza de las últimas N medidas hasta conseguir un resultado lo suficientemente pequeño, lo cual indica que los cambios no están produciendo ya mejoras. También puede determinarse un valor concreto objetivo, como un tiempo de hiperpolarización o fase buscado para imitar el comportamiento de una neurona viva concreta que quiera ser emulada.

Con uno de estos criterios de selección se consigue establecer si la conexión es considerada correcta o debe ser reajustado el parámetro para el cual se esté buscando la mejor configuración posible. Es importante tener en consideración los límites o valores seguros de la variable que esté siendo modificada con el objetivo de reducir el espectro de la búsqueda y proteger el experimento a la vez. Por ejemplo, una corriente inadecuada puede acabar con la neurona viva en la cual sea introducida.

### **5.3.4. Frecuencia de interacción constante e intercambio de información**

Una vez realizado todo el proceso contemplado en el segundo subsistema el programa debe esperar al instante planificado para continuar con su ejecución. Para cumplir este requisito se han utilizado como se ha indicado en el capítulo 3 sistemas operativos con tecnología de tiempo real. En el anexo C se ofrece información sobre los resultados de latencias obtenidos por el programa que implementa el protocolo.

Por otro lado, el intercambio de información que se produce tras cada periodo de espera se maneja con las librerías Comedi para el manejo de tarjetas de adquisición de datos y consiste en la lectura del potencial de membrana de la neurona observada y la inyección en esta de la corriente calculada de acuerdo a la sinapsis producida en el propio programa entre el modelo y la neurona viva.

Los datos producidos así como las señales enviadas son, además, almacenadas durante el proceso de interacción. Para ello, y con el objetivo de conseguir los mejores resultados posibles, así como cumplir con las restricciones temporales, la información a guardar es enviada por una cola de mensajes a otro hilo de ejecución que no tiene prioridad de tiempo real para que sea el encargado de grabar los resultados en un archivo. Esto es debido a que la escritura de los datos en disco es un proceso lento y hacerlo en el mismo hilo de ejecución afecta a las latencias obtenidas. Almacenar correctamente los parámetros de la ejecución así como los datos obtenidos resulta fundamental para su posterior análisis.

Para manejar estas soluciones (Comedi, programación en tiempo real, hilos de ejecución) y permitir definir el experimento se han realizado diferentes módulos utilizados en el programa clamp-cli (ver anexo B).





# 6

## Experimentos y resultados

Este capítulo muestra los resultados de las interacciones híbridas realizadas que han sido realizadas gracias a los protocolos y algoritmos desarrollados. Los resultados muestran que si no se realizan las adaptaciones dinámicas en este proyecto no sería posible establecer interacciones neuronales híbridas eficaces. En las figuras mostradas en este capítulo la señal naranja siempre corresponde a la neurona viva y la azul la neurona software. La frecuencia de muestreo de la señal biológica ha sido la mayor parte del tiempo de 10.000 Hz por ser suficiente, aunque se ha comprobado que en un futuro podría aumentarse a 20.000 Hz. Como resumen el proceso puede ser consultado en la tabla 6.1 y se encuentra detallado en profundidad en los capítulos 4 y 5.

| PASO                           | DESCRIPCIÓN  |
|--------------------------------|--|
| 1 - Calibración temporal       | El valor de integración de la neurona artificial es ajustado para que tenga la misma temporalidad que la neurona viva con las restricciones que impone la tarjeta de adquisición (10-20 kHz).  |
| 2 - Calibración en amplitud    | Se definen variables de escalado en rango y offset en ambos sentidos para que las señales puedan ser expresadas en la escala de la otra neurona. Es decir, cada neurona sigue trabajando en su escala pero puede ser escalada al rango de la otra cuando es necesario. |
| 3 - Calibración de la conexión | Las variables de conductancia y las constantes de tiempo de las conexiones se calibran en función de un objetivo que se evalúa tras una serie de interacciones.  |

Cuadro 6.1: Resumen de los pasos a seguir para conseguir una interacción híbrida.

### 6.1. Análisis y detección de eventos en la señal

---

Para el correcto establecimiento de las conexiones híbridas es necesario adaptar dinámicamente las configuraciones de la conexión y el comportamiento de los modelos. Para esto se requiere analizar la señal y detectar los eventos relevantes de forma precisa. Gracias a estas detecciones de eventos es posible analizar los resultados y modificar los comportamientos generados.

### 6.1.1. Filtrado de la señal

Para obtener mejores resultados y facilitar la detección de eventos en la señal y el análisis online de los datos se han implementado filtros con los cuales eliminar el ruido presente en las señales. Las soluciones implementadas incluyen una convolución (Smith et al., 1997) y una reducción de la dimensionalidad de la señal agrupando valores consecutivos de acuerdo a su media.

La convolución ha sido con lo que mejores valores se han obtenido para su posterior utilización por el resto de algoritmos. El resultado se muestra en las figuras 6.1 y 6.2. En la segunda figura puede observarse como aplicar en mas de una ocasión el filtro sirve para obtener unos resultados más limpios aunque esto a su vez altera el rango de la señal, en concreto de la zona de disparos.

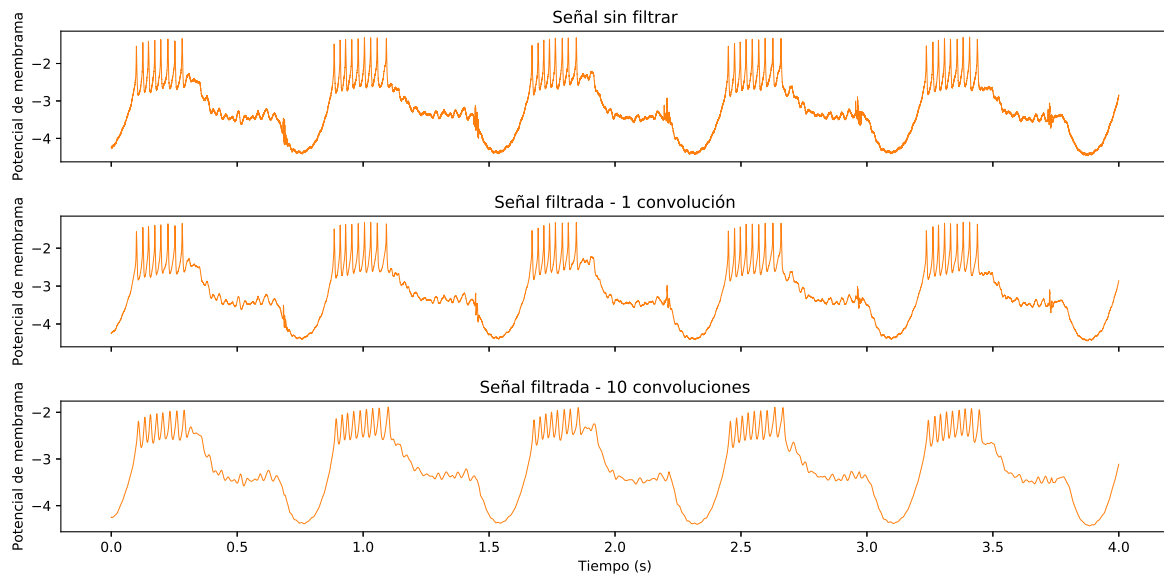


Figura 6.1: Filtrado online de la señal neuronal adquirida.

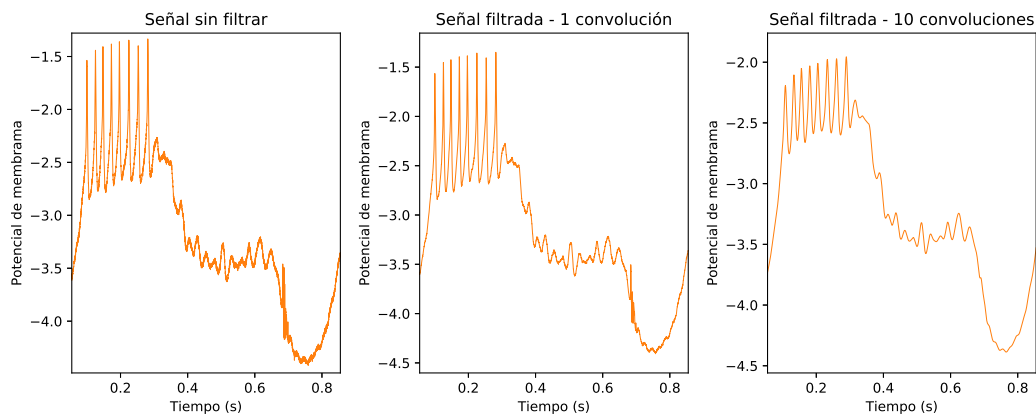


Figura 6.2: Detalle del filtrado online de la señal neuronal adquirida.

Gracias a estos filtrados es posible que algunas de las detecciones de eventos posteriores funcionen de mejor manera. Si en un experimento el ruido en la señal es muy alto también es posible utilizar la señal filtrada para la interacción neuronal y para la medición del nivel de sincronización.

### 6.1.2. Detección de eventos

La detección de eventos como potenciales de acción, ráfagas de disparos, hiperpolarizaciones o despolarizaciones resulta fundamental para caracterizar online la actividad neuronal y que los algoritmos planteados puedan evaluar la situación del sistema. Por ejemplo, en este proyecto ha sido utilizada para definir la frecuencia de la actividad neuronal o para obtener la relación de fase entre dos neuronas. En este caso se ha detectado el inicio de las ráfagas de disparos (figura 6.3), aunque el proceso es fácilmente modificable para detectar el fin de estas o distintos eventos de la señal.

Para ello, en primer lugar fue utilizado un método consistente en un valor frontera de disparo. Sin embargo, esta solución presenta problemas debido a que este valor debe estar situado en un rango de valores muy limitado. Además requiere que el filtrado elimine en esta zona el ruido completamente y los cambios de derivada en la señal.

Para solventar este problema se ha implementado un algoritmo basado en dos valores frontera, uno consistente en la detección del evento, que queda anulado una vez se detecta un evento, y un valor de re-activación de este (algoritmo descrito en el código 6.1).

```
function detectar_rafagas (señal, tiempos, v_deteccion, v_activacion){
    # El estado inicial de la detección se define de acuerdo al primer valor
    up=FALSE
    if (señal[0]>v_activacion)
        up=TRUE

    # Recorremos los valores de la señal
    num_eventos = 0
    valor_eventos = []
    tiempos_eventos = []
    ...
    for (elem, t in señal, tiempos){
        if (up==FALSE and elem>v_deteccion){
            # Evento detectado, se guarda/procesa la información para su
            utilización
            num_eventos++
            valor_eventos.add(elem)
            tiempos_eventos.add(t)
            ...

            # Desactivación de la detección
            up=TRUE
        } else if (up==TRUE and elem<v_activacion){
            # Reactivación de la detección
            up=FALSE
        }
    }
    return num_eventos, valor_eventos, tiempos_eventos, ...
}
```

Código 6.1: Algoritmo para la detección de eventos en una señal neuronal

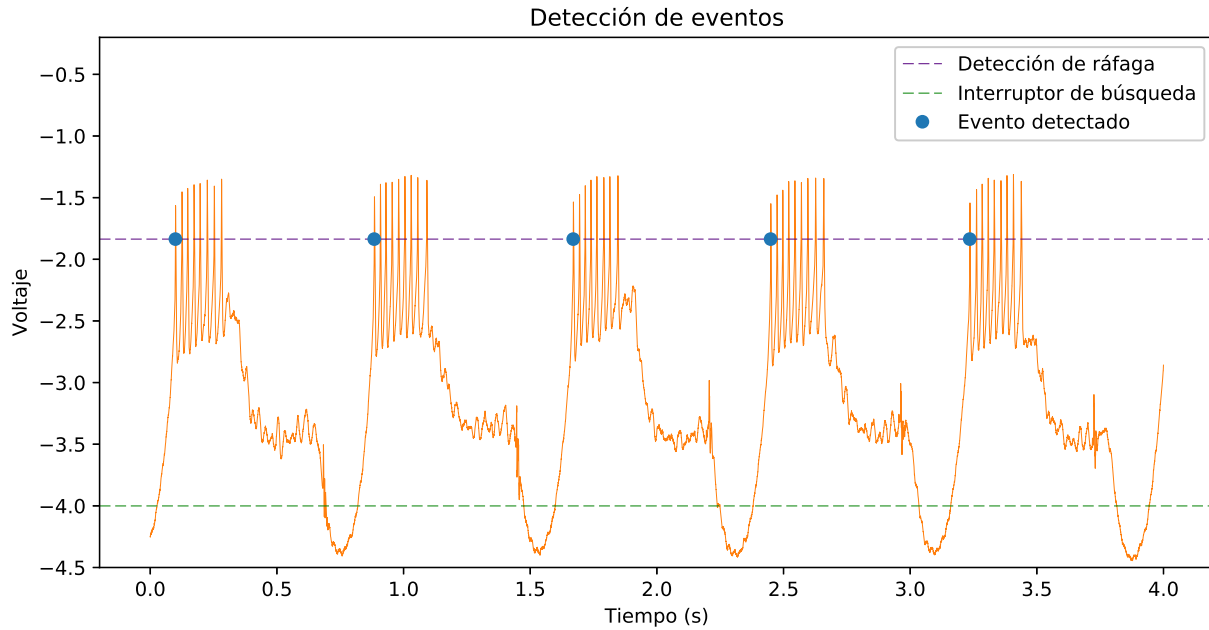


Figura 6.3: Detección del primer disparo de una ráfaga neuronal .

## 6.2. Calibración de escalas de tiempo y de amplitud

Durante el desarrollo (capítulo 5) ya se determinó las diferencias de escalas que deben ser consideradas. Como resumen hay que tener en cuenta en primer lugar que las ráfagas de disparos deben tener una resolución temporal similar siendo si no necesario ajustar los puntos de las ecuaciones del modelo a utilizar (figuras 5.1 y 5.4 y tabla 5.1). Para esto es necesario establecer la frecuencia de disparo de la neurona viva, lo cual es posible gracias a la detección de disparos realizada. Una vez obtenida esta información y conociendo también cuantos puntos de un modelo neuronal producen una ráfaga (para lo cual puede ser también utilizado el algoritmo de análisis o conocer el valor previamente), es posible determinar el número de puntos de integración que serán calculados y cuantos utilizados en la interacción híbrida para conseguir escalar temporalmente de manera correcta los modelos neuronales.

```
function pts_salvar (rafagas_detectadas, t_señal, freq_daq, rafaga_pts_modelo){
    # rafagas_detectadas: determinada por la función detectar_rafagas
    # t_señal: duración de la señal utilizada para determinar rafagas_detectadas
    periodo_disparo = 1 / (rafagas_detectadas / t_observacion)

    # freq_daq: frecuencia de captación de la tarjeta
    rafaga_pts = freq_daq * periodo_disparo

    # rafaga_pts_modelo: definido para cada modelo
    return rafaga_pts_modelo / rafaga_puntos
}
```

Código 6.2: Algoritmo de calibración temporal.

Por otro lado es necesario contar con un proceso que permite adaptar los voltajes de una señal al rango de funcionamiento de otra. Esto es debido a que es necesario que cada neurona reciba la corriente externa en su rango de funcionamiento. Para esto se realiza una observación de la señal biológica, variable en cada experimento, y de los modelos neuronales determinando

su rango de funcionamiento. Una vez hecho esto se calcula el escalado en rango y desplazamiento vertical u offset (en la figura 5.5 se observan dos señales escaladas temporalmente pero que presentan diferente rango o amplitud de voltaje así como offset). A partir de este momento toda señal que sea utilizada en el ámbito de la otra neurona será escalada por medio de los valores calculados. Es decir que, por ejemplo, los modelos de sinapsis son calculados con las unidades del sentido en el cual la corriente va a ser introducida.

```
function escala_v (min_v_modelo, max_v_modelo, min_v_viva, max_v_viva){
    # Cálculo de la amplitud de las señales
    rango_modelo = max_v_modelo - min_v_modelo
    rango_viva = max_v_viva - min_v_viva

    # Calculo para escalar amplitud
    escala_modelo_a_real = rango_viva / rango_virtual
    escala_real_a_modelo = rango_virtual / rango_viva

    # Offset vertical para que los valores coincidan no solo en rango si no
    # también en valores concretos
    offset_modelo_a_real = min_v_viva - (min_v_modelo * escala_modelo_a_real)
    offset_real_a_modelo = min_v_modelo - (min_v_viva * escala_real_a_modelo)

    return escala_modelo_a_real, escala_real_a_modelo, offset_modelo_a_real,
    offset_real_a_modelo
}
```

Código 6.3: Algoritmo de calibración del potencial de membrana.

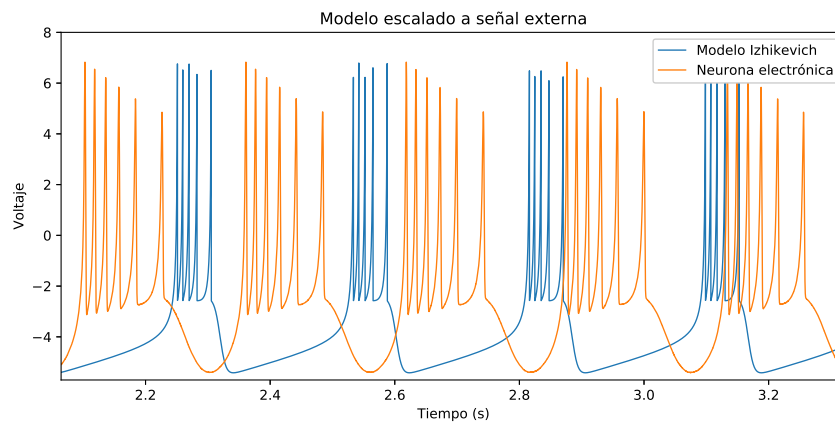


Figura 6.4: Señal neuronal en origen con distinta escala de voltaje y temporalidad adaptada a la dimensión de una neurona electrónica del GNB. Unidades de voltaje Izhikevich =  $[-74, 25; 30, 24]$ . Unidades adaptadas de voltaje Izhikevich =  $[-3, 5; 6, 9]$ . Puntos por ráfaga Izhikevich = 59.324. Puntos por ráfaga adaptados Izhikevich = 2.500.

Una vez realizado este ajuste contamos con un método eficaz de conseguir que el modelo neuronal sea capaz de ser expresado de una manera realista y sin conectarse aún con la neurona viva cuente con temporalidad y rango de funcionamiento similar. En la figura 6.4 se muestra un ejemplo en el cual el modelo de Izhikevich ha sido adaptado para funcionar en la escala espacio temporal de una neurona electrónica. Esto se realiza de manera dinámica basado en las lecturas de la señal. En primer lugar, el modelo ha pasado de una resolución de 59.324 puntos por ráfaga a representarlas en 2.500 (debido a que la tarjeta estaba capturando 10.000 puntos por segundo y la duración de las ráfagas era de 0.25 segundos). Por otro lado este modelo producía valores de potencial de membrana de entre -74 y 30 unidades adimensionales, siendo ahora los mismos valores que la neurona observada.

### 6.3. Sinapsis artificial de dos neuronas externas al programa

Se ha utilizado parte de los algoritmos utilizados, excluyendo lo referente a la parte de simulación de un modelo neuronal, para realizar un circuito entre una neurona electrónica y una neurona viva. La neurona electrónica implementa el modelo Hindmarsh-Rose (Canales, 2017) y para la conexión ha sido utilizada una sinapsis eléctrica (sección 5.1.2).

Los resultados de estos experimentos tienen un límite en la calidad de los resultados debido a que no es posible modificar los valores que determinan el comportamiento de la neurona electrónica, salvo el parámetro  $I$  del modelo Hindmarsh-Rose que regula el comportamiento de los disparos (ver anexo A.3.1). Tampoco puede ser ajustada su temporalidad, por lo que las señales muestran problemas de escala temporal y de rango.

Esto se puede observar en la figura 6.5 si se presta atención a la alta velocidad de disparo de la neurona electrónica respecto a la neurona viva. A pesar de esta limitación en la adaptación la similitud es suficiente para conseguir una conexión. En la figura 6.6 se puede observar la conexión por medio de una sinapsis eléctrica cuando ambas corrientes son positivas. En la figura 6.7 en cambio se observa el resultado cuando una de las corrientes es negativa, lo que provoca una autoinhibición de las neuronas, es decir, que no disparen al mismo tiempo. En adelante se hará referencia a este comportamiento como conexión en antifase o sinapsis inhibitoria.

Estos resultados evidencian las limitaciones de la implementación hardware, que posee una menor posibilidad de variar temporalidad, valores de funcionamiento y comportamiento.

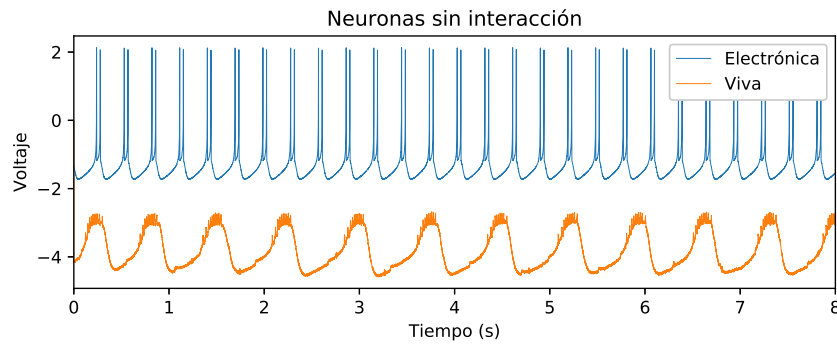


Figura 6.5: Neuronas viva y electrónica sin interacción. Hay diferencia en la frecuencia de disparo.

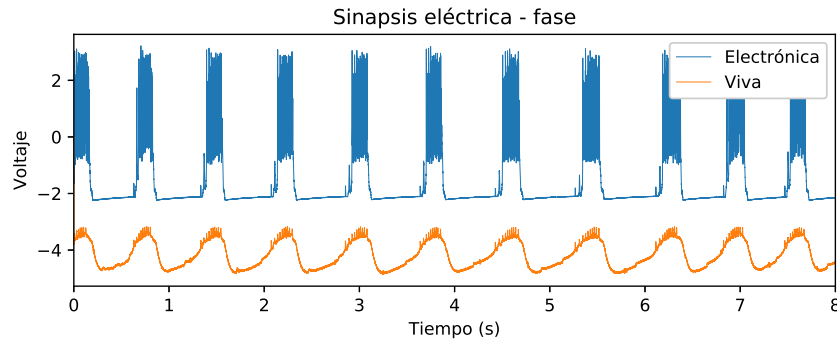


Figura 6.6: Neuronas viva y electrónica conectadas por una sinapsis eléctrica.

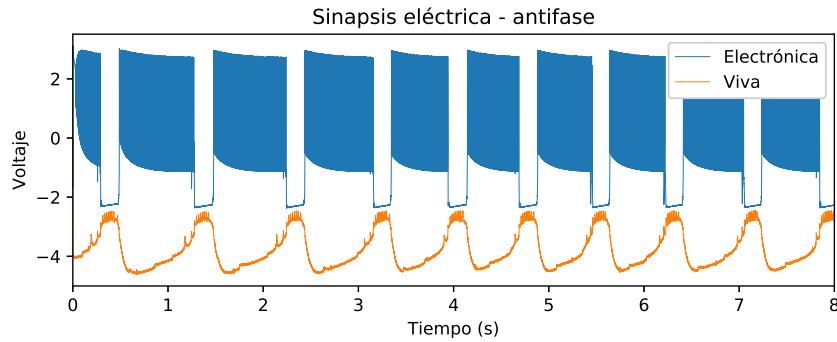


Figura 6.7: Neuronas viva y electrónica conectadas en antifase por una sinapsis eléctrica.

## 6.4. Sinapsis eléctrica

En la figura 6.8 se muestran ejemplo de sinapsis eléctricas (ver capítulo 5.1.2 y anexo A.4.1) producidas entre modelos neuronales y una neurona LP. Como se puede observar la conexión se encuentra completamente escalada y ambas neuronas realizan la conexión sin perder completamente su forma característica propia (a diferencia de la neurona electrónica). Esto es debido a los ajustes descritos anteriormente en este capítulo que no se producen en la neurona electrónica.

La sincronización de las señales neuronales se ilustra en las figuras 6.8, 6.9 y 6.10. La primera figura muestran los potenciales de acción sincronizados en fase (disparo al mismo tiempo) o en antifase (disparos intercalados). La segunda y tercera figura muestran como la relación de potenciales cambia cuando se realiza la sinapsis. Con una conexión en fase la relación se aproxima a una recta  $x = y$  mientras que la conexión en antifase se aproxima a la curva  $y = 1/x$ . Estas relaciones se pueden utilizar para caracterizar los comportamientos de los diferentes modelos de sinapsis y para observar la relación entre ambas señales segmentada entre la hiperpolarización y la despolarización de las señales.

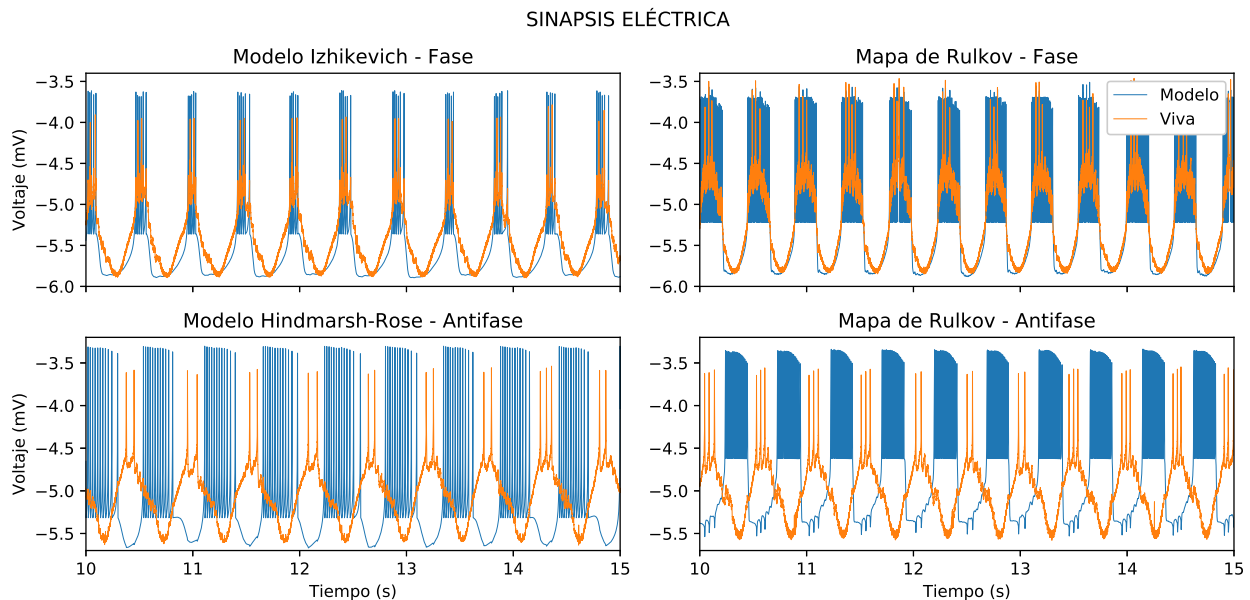


Figura 6.8: Modelos neuronales conectados por sinapsis eléctricas a una neurona viva. Ejemplo de escala (modelo Hindmarsh-Rose): Voltajes originales =  $[-1, 6; 1, 8]$ . Voltajes escalados =  $[-5, 68; -3, 68]$ . Puntos ráfaga = 260.166. Puntos ráfaga escalado = 5.000. Conductancia  $g=0.3$

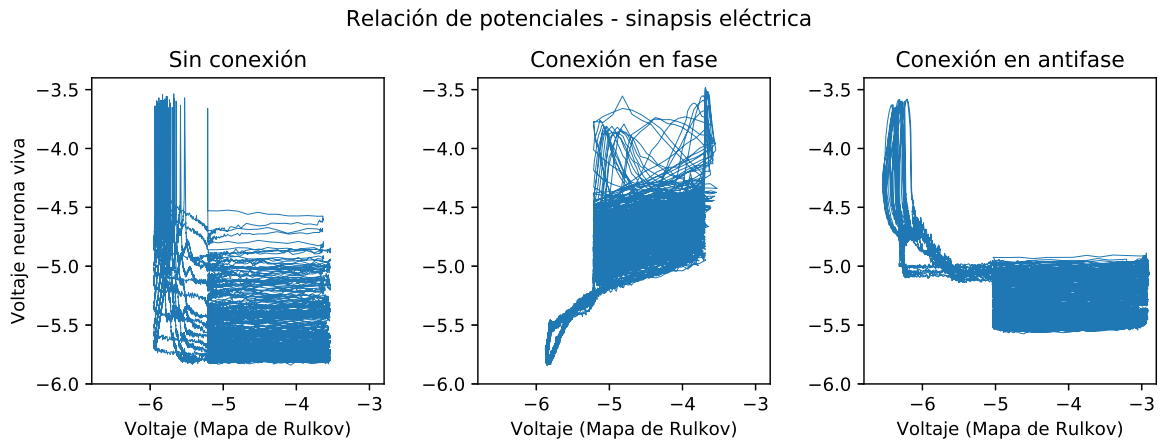


Figura 6.9: Relación entre señales neuronales en una sinapsis eléctrica (Rulkov-LP).

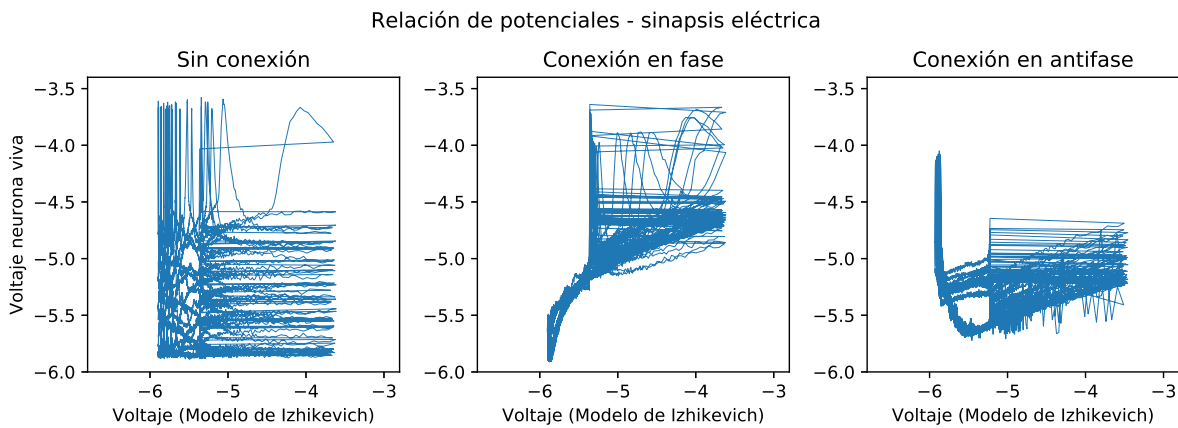


Figura 6.10: Relación entre señales neuronales en una sinapsis eléctrica (Izhikevich-LP).

## 6.5. Sinapsis química

Se ha implementado un modelo de sinapsis química gradual (Golowasch et al., 1999). Este modelo presenta dos posibles conexiones de acuerdo a la velocidad de la sinapsis (ver sección 5.1.2 y anexo A.4.2). En este caso los circuitos híbridos se realizaron con una neurona LP del ganglio estomatogástrico del cangrejo de mar (*Carcinus maenas*). Por este motivo, y para reproducir una conexión presente en el circuito (Harris-Warrick, 1992) se ha realizado una sinapsis química lenta hacia la neurona viva mientras que se ha recibido una sinapsis química rápida (figura 6.11), imitando de esta manera una conexión con una neurona PD.

Como se puede observar en la firma de su relación de potenciales (figura 6.12) esta sinapsis produce una conexión autoinhibitoria correspondiéndose las hiperpolarizaciones de una neurona con las despolarizaciones de la otra y viceversa.



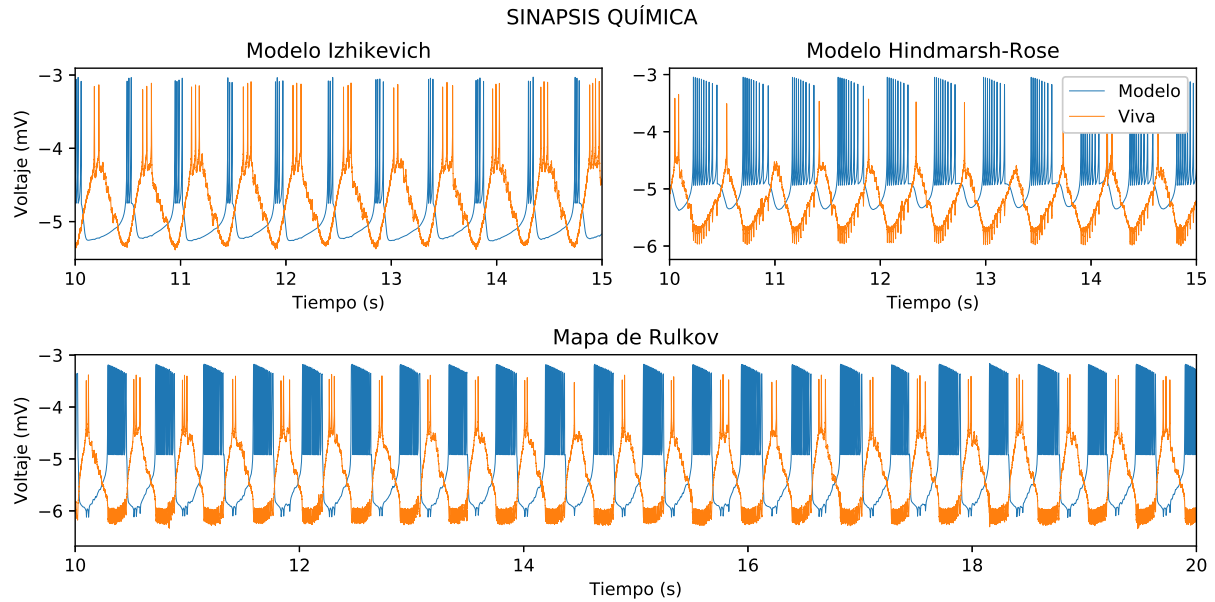


Figura 6.11: Modelos neuronales conectados por sinapsis químicas a una neurona viva. Ejemplo de escala (mapa de Rulkov): Voltajes originales =  $[-1, 9; 2, 1]$ . Voltajes escalados =  $[-6, 2; -3, 2]$ . Puntos ráfaga = 350. Puntos ráfaga escalado = 5.000. Conductancias:  $g_{lenta}=0.02$ ,  $g_{rápida}=0.2$ .

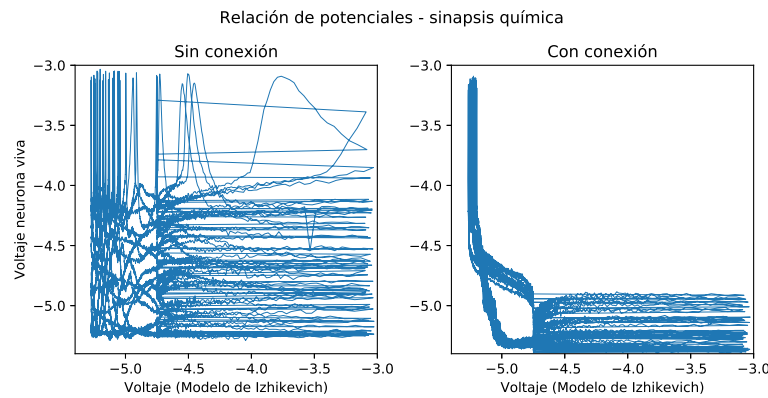


Figura 6.12: Relación entre señales neuronales en una sinapsis química.

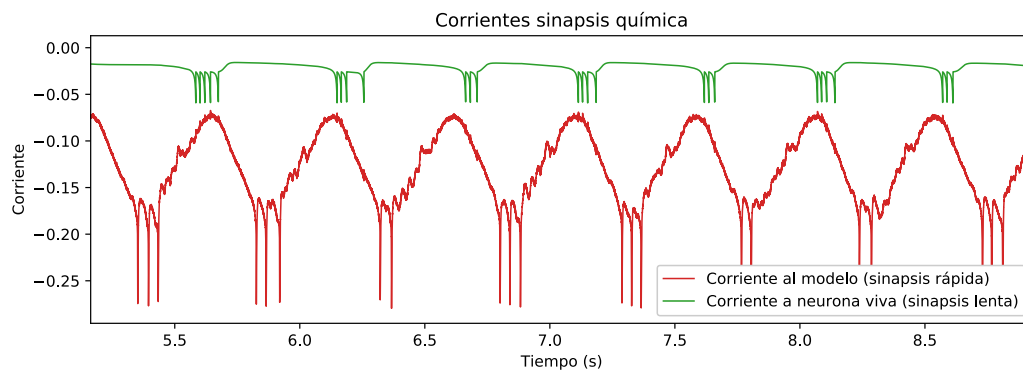


Figura 6.13: Corrientes en una sinapsis químicas.

## 6.6. Calibrado de la conexión en ciclo cerrado

---

Para establecer conexiones híbridas se han realizado procedimientos en ciclo cerrado o barrido de parámetros para determinar de manera dinámica la mejor configuración. Estos procedimientos siguen el protocolo planteado en el capítulo 4 y detallado en el capítulo 5 de esta memoria.

### 6.6.1. Calibración de la conductancia eléctrica

Como ejemplo de los protocolos descritos en los capítulos 4 y 5 se ha realizado una calibración de la conductancia en una sinapsis eléctrica (ver sección 5.1.2). La medida la sincronización utilizada ha sido el error cuadrático medio de la señal. Tras un periodo de observación inicial se ha fijado un valor de conductancia objetivo y se ha realizado una interacción en ciclo cerrado en la cual se aumenta la conductancia de la conexión, observando como afecta esto al sistema. Una vez el requisito de sincronización es alcanzado la interacción continúa con el valor determinado.

En las figuras 6.14 y 6.15 se muestran dos resultados de estos protocolos en los cuales se indica como evoluciona la medida de sincronización y el parámetro que se modifica. Cada figura ha sido obtenida en días de experimentación diferentes y los modelos neuronales son distintos y, a pesar de estos cambios, ambos resultados son correctos.

### 6.6.2. Calibración del comportamiento de los modelos

Alterar el comportamiento del modelo neuronal y comprobar el impacto en el circuito es una manera eficaz de obtener una mejor configuración a la hora de realizar la conexión de acuerdo a un criterio a maximizar o minimizar.

En este caso se ha tratado de conseguir que el modelo Hindmarsh-Rose realice el mismo número de disparos en las ráfagas que la neurona viva. Esto es posible variando el parámetro  $\mu$  en las ecuaciones del modelo (ver anexo A). Como un número de disparos similar aumentará la similitud de las señales en un sinapsis eléctrica en fase el error cuadrático medio puede ser utilizado como métrica en la selección del parámetro.

Se ha realizado este proceso con la neurona electrónica y con una neurona viva (figuras 6.16 y 6.17). Los resultados muestran diferencias pues estos dependen de la neurona observada, aunque en ambos casos se distingue una zona en la cual el error observado se minimiza, lo que indica que valores de  $\mu$  provocan una similitud mayor entre las ráfagas propias y las externas al sistema.

## 6.7. Resumen de los procedimientos realizados

---

Como resumen general de los procedimientos de las secciones anteriores se incluye la figura 6.18. En esta figura se detalla paso a paso los diferentes pasos que deben seguirse para conseguir la construcción correcta de un circuito híbrido.

En primer lugar se observa la neurona viva y el modelo software (modelo de Izhikevich) que se desea conectar. Como se puede ver los potenciales de acción tienen una escala diferente (entre -5.5 y -3.0 mV en el caso de la neurona viva y -80 y 25 unidades adimensionales en el caso del modelo). Además en un mismo periodo de tiempo (1.6 segundos) el número de ráfagas producido ha sido muy distinto. Tras el escalado se puede observar como la temporalidad ha sido igualada y existe la posibilidad de utilizar ambas neuronas en las dos escalas originales.

Tras esto se realizan procedimientos de ciclo cerrado (como se detalla en la sección 6.6 y en las figuras 6.14, 6.15, 6.16 y 6.17) para establecer la conexión entre ambas neuronas.

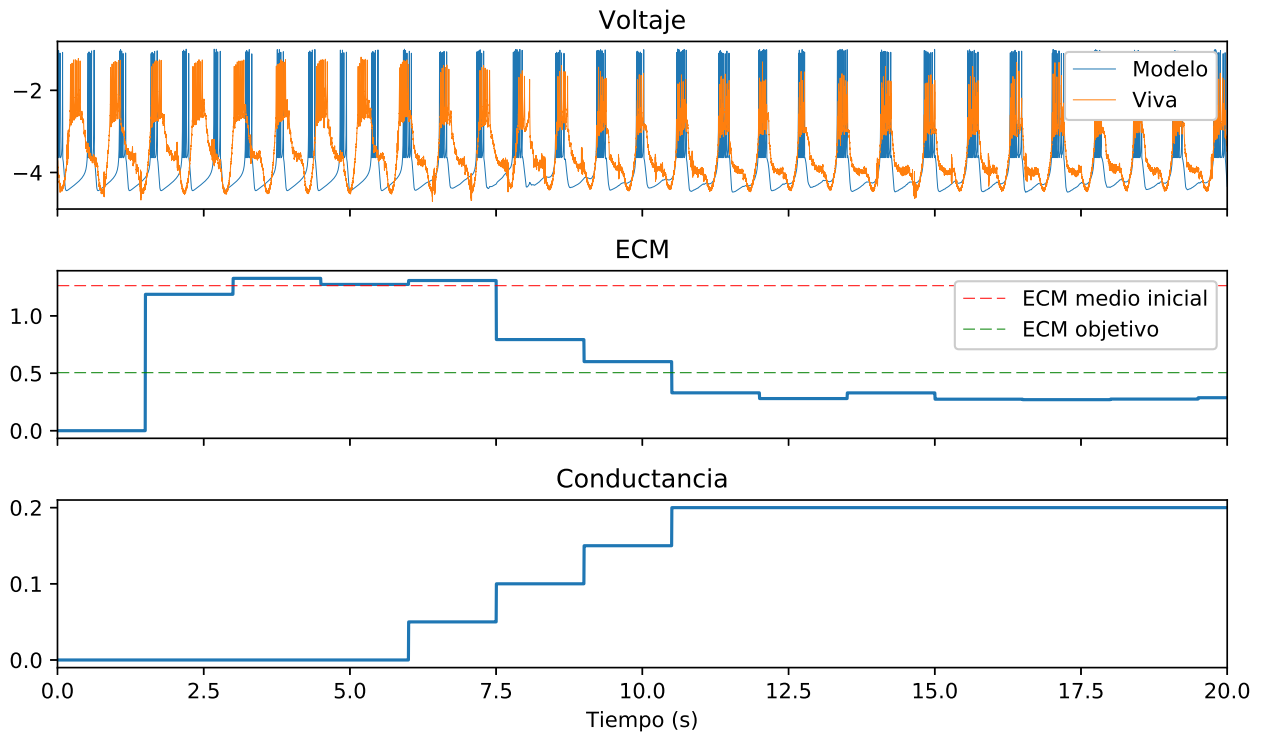


Figura 6.14: Calibración automática de la conductancia de una sinapsis eléctrica por reducción del ECM (40%). Modelo Izhikevich y neurona viva LP (a). Conductancia final  $g = 0,2$ .

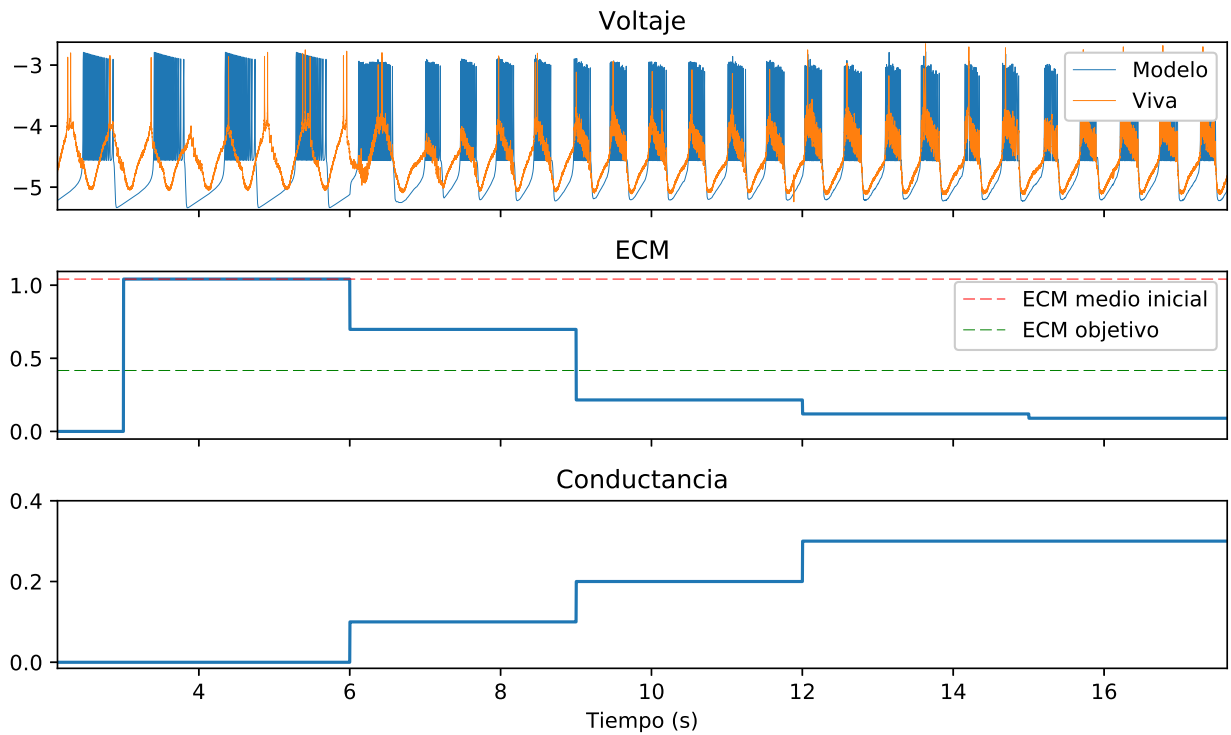


Figura 6.15: Calibración automática de la conductancia de una sinapsis eléctrica por reducción del ECM (40%). Mapa de Rulkov y neurona viva LP (b). Conductancia final  $g = 0,3$ .

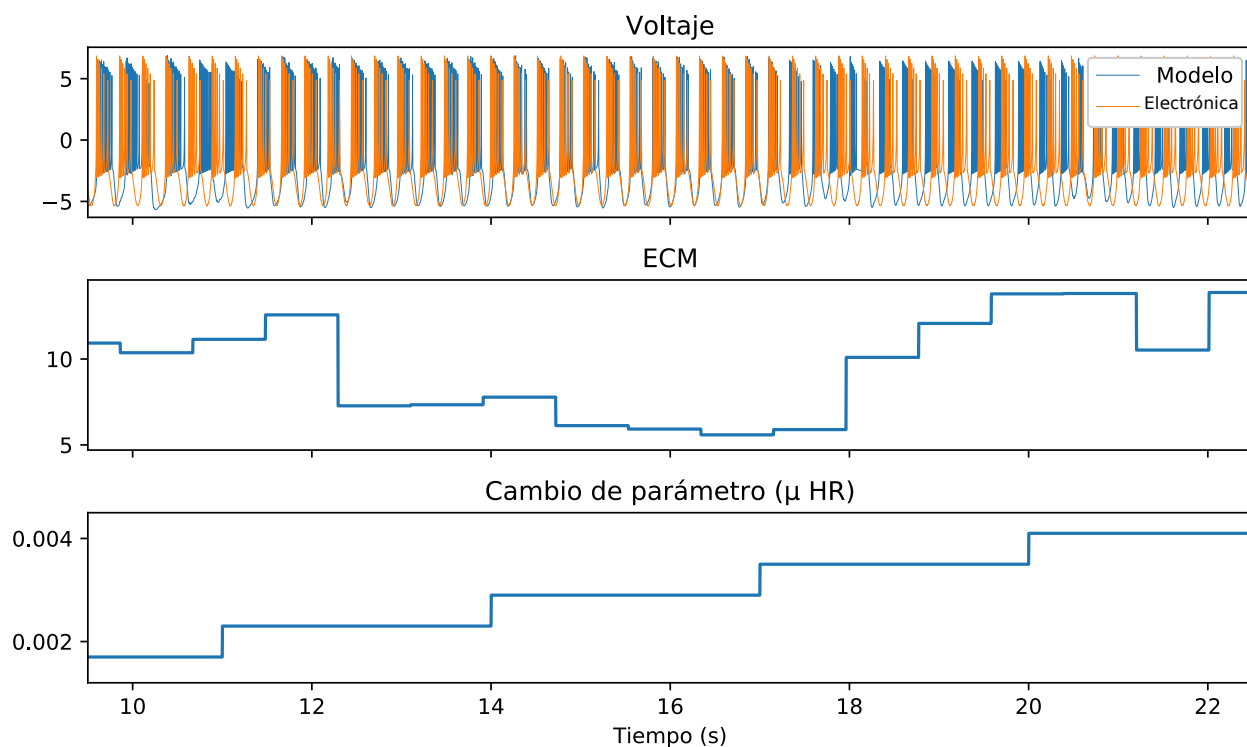


Figura 6.16: Variación del parámetro  $\mu$  del modelo Hindmarsh-Rose. Conexión con una neurona electrónica. Valor óptimo  $\mu = 0,0131$ .

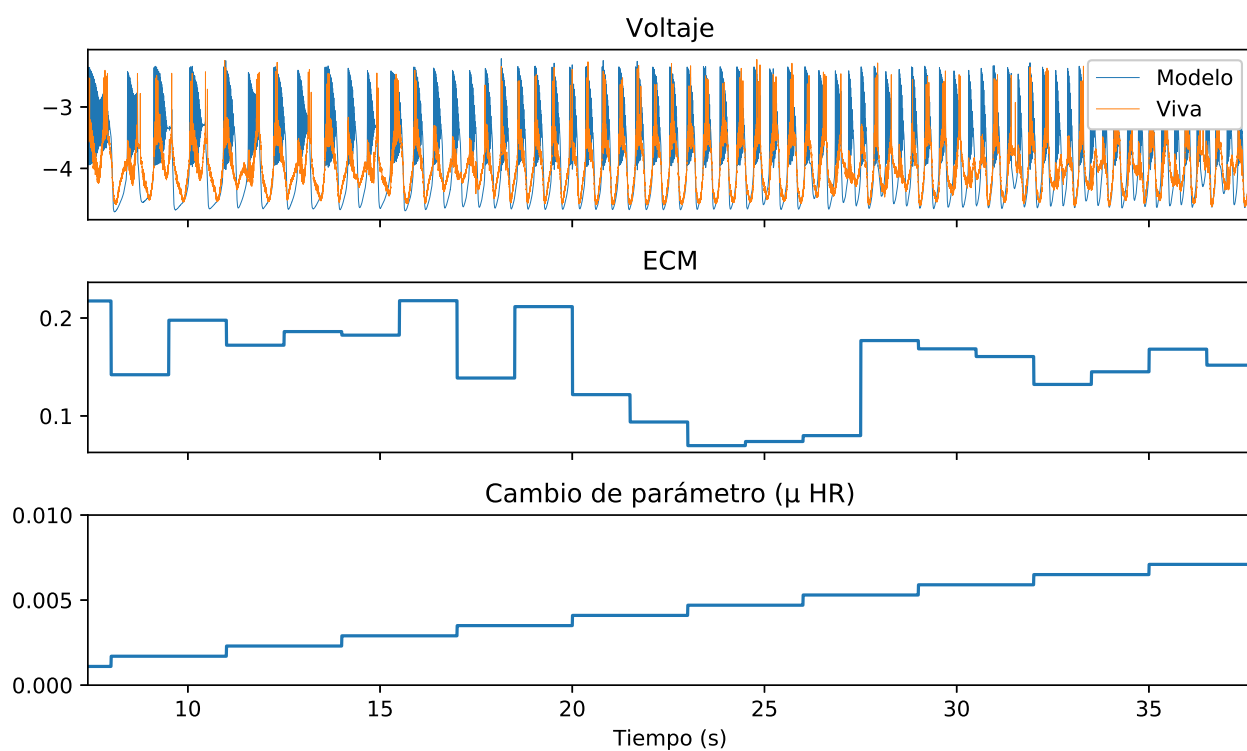


Figura 6.17: Variación del parámetro  $\mu$  del modelo Hindmarsh-Rose. Conexión con una neurona LP viva. Valor óptimo  $\mu = 0,0371$ .

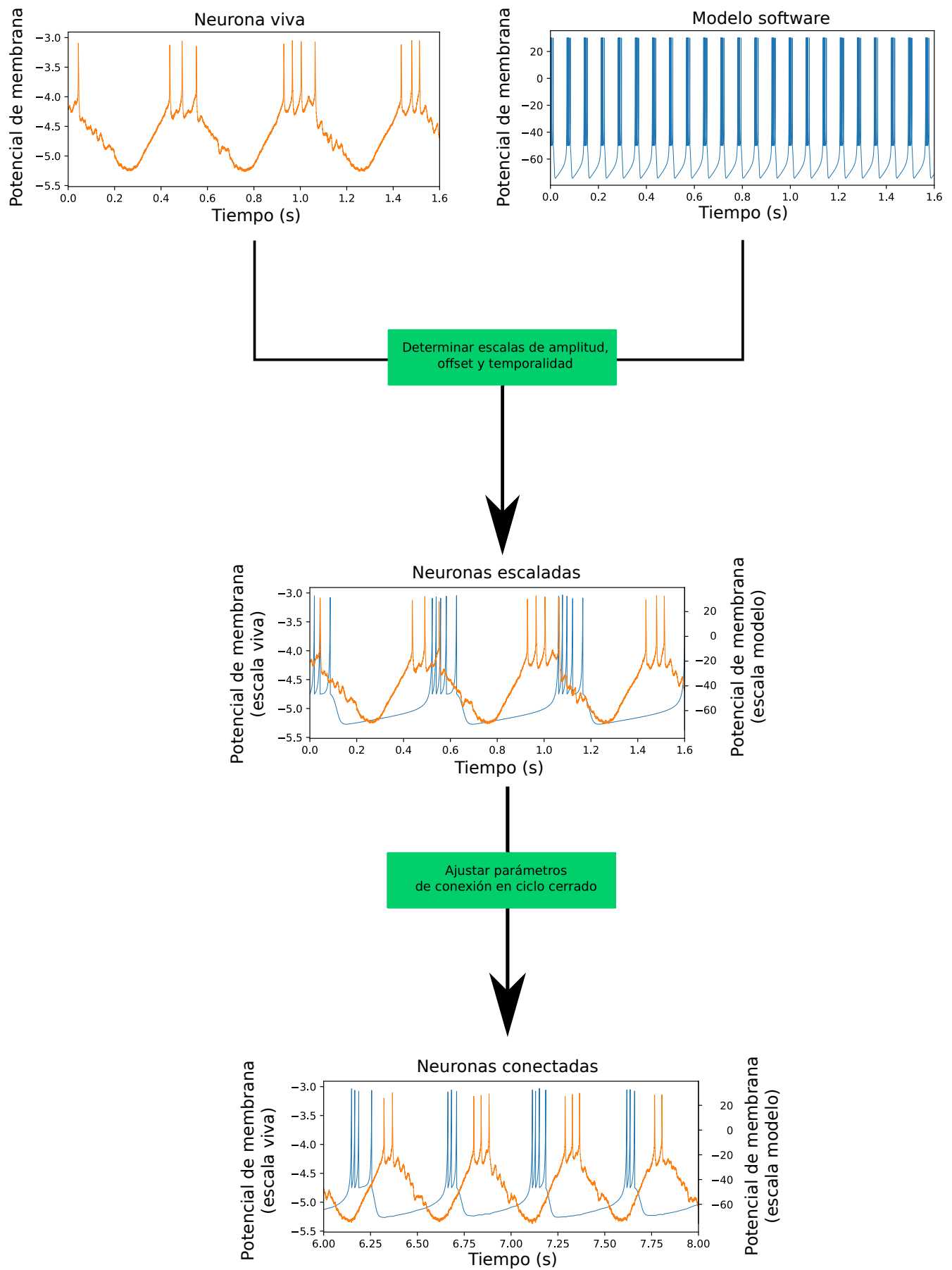


Figura 6.18: Resumen del proceso para la realización de un circuito híbrido.

## 6.8. Mapa de conductancias químicas

Como ejemplo ilustrativo de la automatización de experimentos de circuitos híbridos para la caracterización de dinámicas neuronales se ha realizado una interacción de larga duración en la cual se ha comprobado cómo afecta la variación de las conductancias sinápticas al intervalo desde el primer disparo del modelo al primer disparo de la neurona viva. En este experimento se ha utilizado un circuito híbrido en el cuál la neurona viva ha recibido una sinapsis química lenta mientras que el modelo neuronal (Izhikevich) ha recibido una sinapsis química rápida. Durante el experimento se han barrido las conductancias de las dos conexiones y se ha evaluado para cada par de valores la regularidad y el intervalo temporal entre el primer disparo de las ráfagas de cada neurona. El protocolo realizaba automáticamente la exploración y anotaba esta información.

Estos resultados han sido analizados utilizando el algoritmo de detección de eventos para determinar como afecta la conductancia a la fase entre disparos (figura 6.19). También se ha indicado cuando un par de conductancias no producen una sincronización entre ambas neuronas. Los valores de la conductancia mapeados han sido seleccionados de acuerdo al rango utilizado normalmente para estas sinapsis en simulaciones y publicaciones (Golowasch et al., 1999). Este tipo de mapas se utiliza para caracterizar la dinámica en la negociación de ritmos en los CPG (Elices and Varona, 2017).

El mapa obtenido (figura 6.19) muestra este ejemplo de exploración automática y caracterización de la dinámica de negociación de ritmos en el circuito híbrido. Se observa que la sinapsis lenta por sí sola, o para algunos valores pequeños de la rápida, no consigue establecer un ritmo regular (zonas grises). Se observa también cómo el incremento de la conductancia de la sinapsis rápida, aumenta el intervalo entre el primer disparo de las ráfagas de cada neurona. Para este experimento se realizaron observaciones de 10 segundos para cada par de conductancias, por lo tanto el experimentos se realizó en 10 minutos. Se pueden realizar mapas de estas características para una gran variedad de exploraciones de los parámetros que controlan la negociación del ritmo del circuito híbrido.

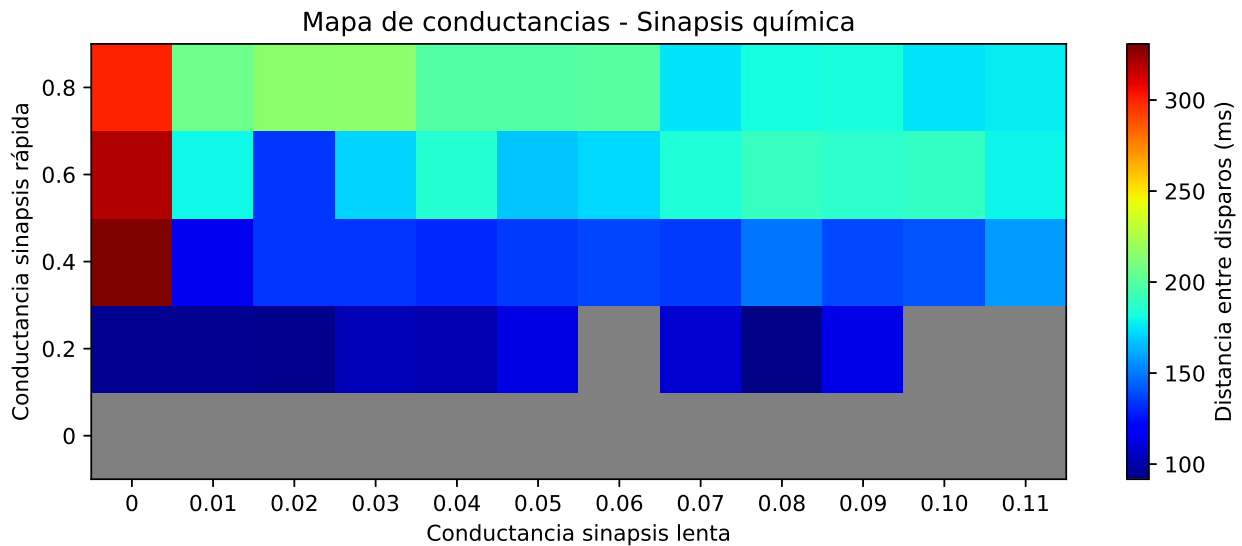


Figura 6.19: Fase generada por diferentes conductancias en una sinapsis química.

# 7

## Conclusiones y trabajo futuro

### 7.1. Conclusiones

---

Los resultados obtenidos en este Trabajo Fin de Máster demuestran que es posible automatizar el ajuste paramétrico requerido para la implementación de circuitos híbridos entre neuronas vivas y modelos neuronales, estableciendo además la metodología para que cumplan objetivos establecidos respecto a la exploración de dinámicas o consecución de un objetivo, consiguiendo interacciones naturales y no forzadas, respetando las dinámicas propias de los sistemas vivos.

Para ello, se han establecido protocolos que definen de manera precisa el proceso y fases a seguir, realizando una descripción detallada de los pasos necesarios para conseguir la realización de estos circuitos. Los procedimientos implementados incluyen un análisis online de las señales para realizar un escalado temporal y de voltajes seguido de un proceso de interacción bidireccional en ciclo cerrado en el cuál es posible evaluar la conexión para adecuar los parámetros que afectan a la conexión y a la simulación de neuronas y sinapsis. Los resultados son dinámicos y están basados en la información obtenida del sistema viva que es cambiante en cada organismo.

El protocolo de interacción y los algoritmos se han implementado teniendo en cuenta que se deben conseguir cumplir las restricciones temporales, del orden de décimas de milisegundos, que imponen las neuronas vivas. Para esto se ha implementado un programa en el lenguaje C, por su eficiencia y control total sobre los recursos, utilizando un sistema operativo de tiempo real y programación orientada a cumplir con las latencias requeridas.

La validación del protocolo y de los algoritmos propuestos ha sido realizada con motoneuronas del ganglio estomatogástrico del cangrejo de mar común. Los resultados han sido satisfactorios y se ha conseguido conectar modelos neuronales con neuronas vivas por medio de sinapsis eléctricas y químicas en diferentes configuraciones. Se ha conseguido producir ritmos de actividad en ráfagas en fase y en antifase, y se ha demostrado la utilidad de estos protocolos para realizar exploraciones automatizadas de la dinámica de este tipo de circuitos.

En conclusión, este trabajo muestra que la realización de circuitos neuronales híbridos es posible pero requiere del uso y conocimiento de tecnologías específicas, así como de la calibración de los comportamientos de los modelos de neuronas y sinapsis. Para ello, es muy positiva la implementación de protocolos de ciclo cerrado que realicen esta calibración de manera automática, presentando este proyecto ejemplos que han sido probados en circuitos híbridos.

## 7.2. Trabajo futuro

---

A partir de los resultados obtenidos se abren múltiples vías de continuación del trabajo a explorar en el futuro.

Por un lado, es posible seguir incrementando los modelos de sinapsis y neuronas disponibles en la plataforma, los métodos de calibración de las conexiones o portar el trabajo a otras plataformas de tiempo real (RTAI, Xenomai). También sería deseable incluir el protocolo y los algoritmos en una plataforma más sencilla de utilizar con interfaz gráfica, como RTBiomanager, la plataforma desarrollada por el GNB para el establecimiento de circuitos cerrados en distintas escalas de descripción del sistema nervioso (Muñiz et al., 2009).

Por otro lado, las técnicas descritas en este trabajo pueden ser utilizadas para comprobar hipótesis sobre las dinámicas de los sistemas neuronales y servir para comprender el funcionamiento de estos, así como para la automatización de experimentos o realizarlos de manera más precisa y exacta. Además pueden servir para construir modelos neuronales que reproduzcan todos los aspectos de la dinámica neuronal revelados por la interacción bidireccional.

También es posible plantear un aumento de las características disponibles en el protocolo, simulando más de una neurona, lo que permitiría realizar circuitos de modelos neuronales con una neurona viva y el resto artificiales. Esto podría ser utilizado, por ejemplo, para realizar un robot con un movimiento regular dirigido por un CPG y que gracias al feedback introducido en el sistema por medio de las neuronas artificiales se aprovecharía de la flexibilidad para negociar ritmos de los circuitos generadores centrales de patrones.

Cabe la posibilidad también de generalizar los algoritmos y protocolos a otros tipos de técnicas experimentales de neurociencia como interfaces cerebro-máquina (Fernandez-Vargas et al., 2013), experimentos del comportamiento en animales (Lareo et al., 2017), real-time fMRI (Rana et al., 2016) o experimentos donde se monitoriza y actúa con señales de distinta naturaleza (Muñiz et al., 2008).

Los algoritmos desarrollados y los resultados obtenidos serán difundidos en congresos científicos (ver anexo D) y próximamente se publicarán los resultados.



# Bibliografía

- Amaducci, R. (2017). Real-time software technology and its use in experimental neuroscience. M.S. thesis, Universidad Autónoma de Madrid.
- Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Beal, D., Bianchi, E., Dozio, L., Hughes, S., Mantegazza, P., and Papacharalambous, S. (2000). Rtai: Real-time application interface. *Linux Journal*, 29(10).
- Biró, I. and Giugliano, M. (2015). A reconfigurable visual-programming library for real-time closed-loop cellular electrophysiology. *Frontiers in neuroinformatics*, 9.
- Broccard, F. D., Joshi, S., Wang, J., and Cauwenberghs, G. (2017). Neuromorphic neural interfaces: from neurophysiological inspiration to biohybrid coupling with nervous systems. *Journal of neural engineering*, 14(4):041002.
- Canales, N. (2017). Implementación y acondicionamiento de neuronas en pcbs. B.S. thesis, Universidad Autónoma de Madrid.
- Chamorro, P., Levi, R., Rodríguez, F. B., Pinto, R. D., and Varona, P. (2009). Real-time activity-dependent drug microinjection. *BMC Neuroscience*, 10(1):P296.
- Chamorro, P., Muñiz, C., Levi, R., Arroyo, D., Rodríguez, F., and Varona, P. (2012). Generalization of the dynamic clamp concept in neurophysiology and behavior. *PLoS ONE*, 7.
- Cole, K. S. and Curtis, H. J. (1939). Electric impedance of the squid giant axon during activity. *The Journal of general physiology*, 22:649–670.
- Cole, K. S. and Marmont, G. (1942). The effect of ionic environment upon the longitudinal impedance of the squid giant axon. In *Fed. Proc*, volume 1, pages 15–16.
- Connor, J. A., Walter, D., and McKown, R. (1977). Neural repetitive firing: modifications of the hodgkin-huxley axon suggested by experimental results from crustacean axons. *Biophysical journal*, 18:81–102.
- Destexhe, A. and Bal, T. (2009). *Dynamic Clamp: From Principles to Applications*. Springer.
- Destexhe, A., Mainen, Z. F., and Sejnowski, T. J. (1994). An efficient method for computing synaptic conductances based on a kinetic model of receptor binding. *Neural computation*, 6(1):14–18.
- Elices, I. and Varona, P. (2015). Closed-loop control of a minimal central pattern generator network. *Neurocomputing*, 170:55–62.
- Elices, I. and Varona, P. (2017). Asymmetry factors shaping regular and irregular bursting rhythms in central pattern generators. *Frontiers in computational neuroscience*, 11.

- Elson, R. C., Selverston, A. I., Huerta, R., Rulkov, N. F., Rabinovich, M. I., and Abarbanel, H. D. (1998). Synchronous behavior of two coupled biological neurons. *Physical Review Letters*, 81(25):5692.
- Fernandez-Vargas, J., Pfaff, H. U., Rodríguez, F. B., and Varona, P. (2013). Assisted closed-loop optimization of ssvep-bci efficiency.
- Gerum, P. (2004). Xenomai-implementing a rtos emulation framework on gnu/linux. *White Paper, Xenomai*, page 81.
- Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256.
- Golowasch, J., Casey, M., Abbott, L., and Marder, E. (1999). Network stability from activity-dependent regulation of neuronal conductances. *Neural Computation*, 11(5):1079–1096.
- Gómez Nicola, D. and Nieto Sampedro, M. (2008). Glía reactiva. *Mente y Cerebro*, 32:78–87.
- Harris-Warrick, R. M. (1992). *Dynamic biological networks: the stomatogastric nervous system*. MIT press.
- Harris-Warrick, R. M. (2011). Neuromodulation and flexibility in central pattern generator networks. *Current Opinion in Neurobiology*, 21:685–692.
- Hebb, D. O. (1949). *The Organization of Behaviour*. Organization.
- Herrero-Carrón, F., Rodríguez, F. B., and Varona, P. (2011). Bio-inspired design strategies for central pattern generator control in modular robotics. *Bioinspiration & biomimetics*, 6.
- Hille, B. (1977). Ionic channels in excitable membranes. *Biophysical Journal*, 22:283–294.
- Hindmarsh, J. and Rose, R. (1984). A model of neuronal bursting using three coupled first order differential equations. *Proceedings of the Royal Society of London. Series B, Biological sciences*, 221:87–102.
- Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its applicaiton to conduction and excitation in nerve. *J Physiol*, 117:500–544.
- Hooper, R. M., Tikidji-Hamburyan, R. A., Canavier, C. C., and Prinz, A. A. (2015). Feed-back control of variability in the cycle period of a central pattern generator. *Journal of neurophysiology*, 114(5):2741–2752.
- Hormuzdi, S. G., Filippov, M. A., Mitropoulou, G., Monyer, H., and Bruzzone, R. (2004). Electrical synapses: a dynamic signaling system that shapes the activity of neuronal networks. *Biochimica et Biophysica Acta (BBA) - Biomembranes*, 1662:113137.
- Hull, T., Enright, W., Fellen, B., and Sedgwick, A. (1972). Comparing numerical methods for ordinary differential equations. *SIAM Journal on Numerical Analysis*, 9(4):603–637.
- Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14:1569–1572.
- Kandel, E. R., Schwartz, J. H., and Jessell, T. M. (2000). Principles of neural science. *Neurology*, 4.
- Kemenes, I., Marra, V., Crossley, M., Samu, D., Staras, K., Kemenes, G., and Nowotny, T. (2011). Dynamic clamp with stdpc software. *Nature protocols*, 6(3):405–417.

- Kepler, T. B., Abbott, L. F., and Marder, E. (1992). Reduction of conductance-based neuron models. *Biological Cybernetics*, 66:381–387.
- Lareo, Á., Forlim, C. G., Pinto, R. D., Varona, P., and Rodríguez, F. B. (2017). Analysis of electroreception with temporal code-driven stimulation. In *International Work-Conference on Artificial Neural Networks*, pages 101–111. Springer.
- Le Masson, G., Renaud-Le Masson, S., Debay, D., and Bal, T. (2002). Feedback inhibition controls spike transfer in hybrid thalamic circuits. *Nature*, 417(6891):854–858.
- Liao, Q., Zhou, S., Shi, H., and Shi, W. (2017). Parameter estimation of nonlinear systems by dynamic cuckoo search. *Neural Computation*, 29(4):1103–1123.
- Linaro, D., Couto, J., and Giugliano, M. (2014). Command-line cellular electrophysiology for conventional and real-time closed-loop experiments. *Journal of neuroscience methods*, 230:5–19.
- Linaro, D., Couto, J., and Giugliano, M. (2015). Real-time electrophysiology: Using closed-loop protocols to probe neuronal dynamics and beyond. *JoVE (Journal of Visualized Experiments)*, pages e52320–e52320.
- López-Muñoz, F., Boya, J., and Alamo, C. (2006). Neuron theory, the cornerstone of neuroscience, on the centenary of the nobel prize award to santiago ramón y cajal. *Brain Res Bull*, 70:391405.
- Mcculloch, W. S. and Pitts, W. (1943). A logical calculus of the idea immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133.
- Morris, C. and Lecar, H. (1981). Voltage oscillations in the barnacle giant muscle fiber. *Biophysical journal*, 35:193–213.
- Mulansky, M., Bozanic, N., Sburlea, A., and Kreuz, T. (2015). A guide to time-resolved and parameter-free measures of spike train synchrony. In *Event-based Control, Communication, and Signal Processing (EBCCSP), 2015 International Conference on*, pages 1–8. IEEE.
- Muñiz, C., Levi, R., Benkrid, M., Rodríguez, F. B., and Varona, P. (2008). Real-time control of stepper motors for mechano-sensory stimulation. *Journal of neuroscience methods*, 172(1):105–111.
- Muñiz, C., Rodríguez, F. B., and Varona, P. (2009). Rtbio manager: a software platform to expand the applications of real-time technology in neuroscience. *BMC Neurosci*, 10(Suppl 1):P49.
- Nagumo, J., Arimoto, S., and Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50:2061–2070.
- Netter, F. (1991). Sistema nervioso: anatomía y fisiología (tomo 1.1).
- Nowotny, T. and Varona, P. (2012). Dynamic clamp. *Encyclopedia of Nanotechnology*, pages 613–621.
- Nowotny, T., Zhigulin, V. P., Selverston, A. I., Abarbanel, H. D., and Rabinovich, M. I. (2003). Enhancement of synchronization in a hybrid neural circuit by spike-timing dependent plasticity. *Journal of Neuroscience*, 23(30):9776–9785.
- Oliveira, D. B. and Oliveira, R. S. (2015). Timing analysis of the preempt rt linux kernel. *Software: Practice and Experience*.

- Oprisan, S., Prinz, A., and Canavier, C. (2004). Phase resetting and phase locking in hybrid circuits of one model and one biological neuron. *Biophysical journal*, 87(4):2283–2298.
- Ortega, F. A., Butera, R. J., Christini, D. J., White, J. A., and Dorval, A. D. (2014). Dynamic clamp in cardiac and neuronal systems using rtxi. *Patch-Clamp Methods and Protocols*, pages 327–354.
- Pinto, R. D., Varona, P., Volkovskii, A., Szücs, A., Abarbanel, H. D., and Rabinovich, M. I. (2000). Synchronous behavior of two coupled electronic neurons. *Physical Review E*, 62(2):2644.
- Prinz, A. A., Abbott, L., and Marder, E. (2004). The dynamic clamp comes of age. *Trends in neurosciences*, 27(4):218–224.
- Rabinovich, M. I., Varona, P., Selverston, A. I., and Abarbanel, H. D. I. (2006). Dynamical principles in neuroscience. *Reviews of Modern Physics*, 78.
- Rana, M., Varan, A. Q., Davoudi, A., Cohen, R. A., Sitaram, R., and Ebner, N. C. (2016). Real-time fmri in neuroscience research and its use in studying the aging brain. *Frontiers in Aging Neuroscience*, 8.
- Reyes-Sánchez, M. (2016). Calibración automática de modelos neuronales para su uso en circuitos híbridos. B.S. thesis, Universidad Autónoma de Madrid.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.
- Roth, E., Sponberg, S., and Cowan, N. J. (2014). A comparative approach to closed-loop computation. *Current Opinion in Neurobiology*, 25:54–62.
- Rulkov, N. F. (2002). Modeling of spiking-bursting neural behavior using two-dimensional map. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 65.
- Schiff, S. J. (2010). Towards model-based control of parkinson’s disease. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 368(1918):2269–2308.
- Selverston, A. I., Rabinovich, M. I., Abarbanel, H. D., Elson, R., Szücs, A., Pinto, R. D., Huerta, R., and Varona, P. (2000). Reliable circuits from irregular neurons: a dynamical approach to understanding central pattern generators. *Journal of Physiology-Paris*, 94(5):357–374.
- Sharp, A. A., Abbott, L., and Marder, E. (1992). Artificial electrical synapses in oscillatory networks. *Journal of Neurophysiology*, 67(6):1691–1694.
- Skinner, F. K., Turrigiano, G. G., and Marder, E. (1993). Frequency and burst duration in oscillating neurons and two-cell networks. *Biological Cybernetics*, 69:375–383.
- Smith, S. W. et al. (1997). *The scientist and engineer’s guide to digital signal processing*. California Technical Pub. San Diego.
- Stigen, T., Danzl, P., Moehlis, J., and Netoff, T. (2011). Controlling spike timing and synchrony in oscillatory neurons. *Journal of neurophysiology*, 105(5):2074.
- Szücs, A., Varona, P., Volkovskii, A. R., Abarbanel, H. D., Rabinovich, M. I., and Selverston, A. I. (2000). Interacting biological and electronic neurons generate realistic oscillatory rhythms. *Neuroreport*, 11(3):563–569.

- Tiesinga, P., Fellous, J.-M., and Sejnowski, T. J. (2008). Regulation of spike timing in visual cortical circuits. *Nature reviews neuroscience*, 9(2):97–107.
- Torres, J. J. and Varona, P. (2012). Modeling biological neural networks. In *Handbook of Natural Computing*, pages 533–564. Springer.
- Tsodyks, M. V. and Markram, H. (1997). The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *Proceedings of the National Academy of Sciences*, 94(2):719–723.
- Varona, P., Torres, J. J., Abarbanel, H. D., Rabinovich, M. I., and Elson, R. C. (2001a). Dynamics of two electrically coupled chaotic neurons: experimental observations and model analysis. *Biological Cybernetics*, 84(2):91–101.
- Varona, P., Torres, J. J., Huerta, R., Abarbanel, H. D., and Rabinovich, M. I. (2001b). Regularization mechanisms of spiking–bursting neurons. *Neural Networks*, 14(6):865–875.
- Yarom, Y. (1991). Rhythmogenesis in a hybrid system—interconnecting an olivary neuron to an analog network of coupled oscillators. *Neuroscience*, 44(2):263–275.



## ANEXOS







## Modelos neuronales y de sinapsis

### A.1. Integración de ecuaciones diferenciales ordinarias en tiempo real

---

Dos de los usados en este trabajo, así como buena parte de los modelos neuronales utilizan ecuaciones diferenciales ordinarias (EDO). Para resolver este tipo de ecuaciones se incluyen a continuación dos métodos de integración numérica que pueden ser utilizados.

También cabe destacar que estos elementos requieren de selección de un paso de integración adecuado. Un paso muy grande provoca señales incorrectas mientras que uno demasiado grande sobrecarga el sistema, pues se alteran los cálculos necesario entre interacciones (Reyes-Sánchez, 2016). Además el cambio de este parámetro afecta a los puntos por ráfaga de un modelo, algo que tienen en cuenta los algoritmos diseñados en este TFM.

#### A.1.1. Método de Euler

Dado que una ecuación diferencial por definición es aquella que relaciona la propia función con su derivada es posible definirla como un límite.

$$\frac{df(t)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{f(t + \Delta t) - f(t)}{\Delta t}$$

Con un incremento del tiempo próximo a cero es posible eliminar el límite

$$\frac{df(t)}{dt} = \frac{f(t + \Delta t) - f(t)}{\Delta t}$$

Finalmente despejamos el próximo instante de tiempo, ya que es el valor buscado

$$f(t + \Delta t) = f(t) + \Delta t \frac{df(t)}{dt}$$

### A.1.2. Método de Runge-Kutta

Aunque el método anterior ofrece un rendimiento suficiente para los modelos utilizados en este trabajo puede ser necesario una mayor rapidez de integración si se utilizan modelos neuronales con mayor carga computacional o se requiere simular más de una neurona. Por este motivo es posible utilizar un método de Runge-Kutta, que mejora la precisión respecto al método de Euler con pasos de integración menores. En este caso se ha utilizado un método de orden 6 (Hull et al., 1972). La implementación se encuentra detallada en:

[www.github.com/manurs/clamp-cli/blob/master/model\\_library.c](https://www.github.com/manurs/clamp-cli/blob/master/model_library.c)

## A.2. Interpolación lineal

---

Si la resolución de un modelo es menor que la que se obtiene de la neurona biológica (es decir, que un evento como una ráfaga de disparos se expresa en menos puntos que el mismo evento en la neurona viva) es necesario aumentar la resolución del resultado del modelo. Para ello en este trabajo se ha realizado una interpolación lineal.

La interpolación lineal consiste en producir nuevos valores entre dos puntos de acuerdo a la recta que une ambos puntos, generando con una distancia uniforme tantos puntos como sea necesario.

Función de interpolación utilizada:

$$f(x|x_1; x_2) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1)$$

## A.3. Modelos neuronales

---

A continuación se ofrece información sobre los modelos neuronales utilizados, incluyendo los parámetros iniciales seleccionados o valores de convergencia. En la tabla 5.1 ya se indicó el número de puntos por ráfaga que generan los modelos para el paso de integración utilizado.

### A.3.1. Hindmarsh-Rose

Potencial de membrana:

$$\frac{dx(t)}{dt} = ay(t) + bx(t)^2 - cx(t)^3 - dz(t) + I$$

Dinámica de los canales iónicos rápidos:

$$\frac{dy(t)}{dt} = e - fx(t)^2 - y(t)$$

Dinámica de los canales iónicos lentos:

$$\frac{dz(t)}{dt} = \mu(-z(t) + S(x(t) + h))$$

El modelo Hindmarsh-Rose puede producir un comportamiento caótico en la duración de las ráfagas si se cumple que  $I > 3.281$

| Variable | Valor     |
|----------|-----------|
| x        | -0.712841 |
| y        | -1.936878 |
| z        | 3.165682  |

Cuadro A.1: Valores iniciales para el modelo Hindmarsh-Rose

| Parámetro | Valor  |
|-----------|--------|
| a         | 1.0    |
| b         | 3.0    |
| c         | 1.0    |
| z         | 1.0    |
| I         | 3.0    |
| e         | 1.0    |
| f         | 5.0    |
| $\mu$     | 0.0021 |
| S         | 4.0    |
| h         | 1.6    |

Cuadro A.2: Parámetros seleccionados inicialmente para el modelo Hindmarsh-Rose

### A.3.2. Izhikevich

Potencial de membrana:

$$\frac{dv(t)}{dt} = 0,004v^2 + 5v + 140 - u + I$$

Variable de recuperación de membrana:

$$\frac{du(t)}{dt} = a(bv - u)$$

Valor de reset:

$$v \geq 30 \Rightarrow \begin{cases} v = c \\ u = u + d \end{cases}$$

| Variable | Valor     |
|----------|-----------|
| v        | 30.240263 |
| u        | -5.544592 |

Cuadro A.3: Valores iniciales para el modelo Izhikevich

| Parámetro | Valor |
|-----------|-------|
| I         | 10.0  |
| a         | 0.02  |
| b         | 0.2   |
| c         | -50.0 |
| d         | 2.0   |

Cuadro A.4: Parámetros seleccionados para el modelo Izhikevich

Este modelo dispone de distintos modos de funcionamiento que son indicados en la figura A.1 que el autor ofrece para su reproducción en [www.izhikevich.org/publications/spikes.htm](http://www.izhikevich.org/publications/spikes.htm)

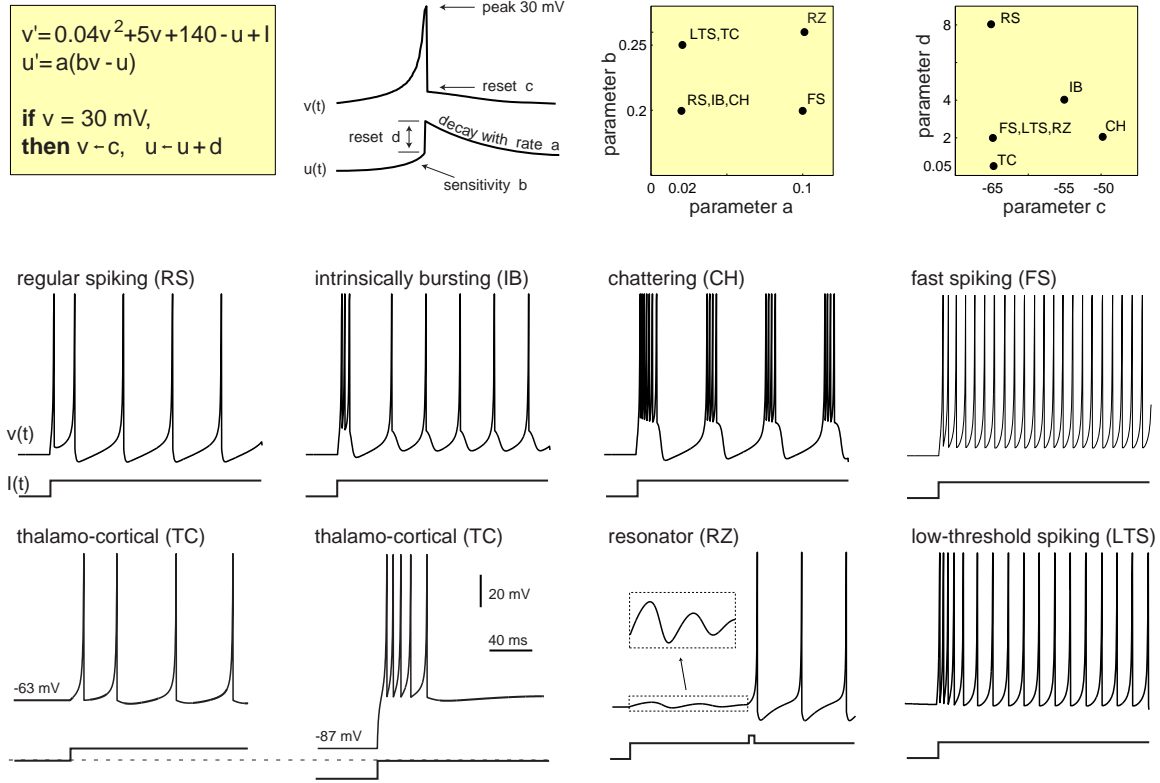


Figura A.1: Mapa de selección de los parámetros del modelo de Izhikevich. Figura reproducida de [www.izhikevich.org/publications/spikes.htm](http://www.izhikevich.org/publications/spikes.htm)

### A.3.3. Mapa de Rulkov

Potencial de membrana:

$$x_{n+1} = f(x_n, y_n + eI_n)$$

$$f(x, y) = \begin{cases} \frac{\alpha}{1-x} + y & \text{si } x \leq 0 \\ \alpha + y & \text{si } 0 < x < \alpha + y \\ -1 & \text{si } x \geq \alpha + y \end{cases}$$

Variable de dinámica lenta:

$$y_{n+1} = y_n - \mu(x_n + 1) + \mu\sigma$$

Este modelo presenta tres comportamientos: silencio, disparo y ráfaga de disparos. Este es definido por dos variables (ver figura A.2).

| Variable | Valor     |
|----------|-----------|
| x        | -1.701747 |
| y        | -4.919226 |

Cuadro A.5: Valores iniciales para el mapa de Rulkov

| Parámetro | Valor |
|-----------|-------|
| $\alpha$  | 6.0   |
| $\mu$     | 0.001 |
| $\sigma$  | -.01  |
| e         | 1     |
| I         | 1     |

Cuadro A.6: Parámetros seleccionados inicialmente para el mapa de Rulkov

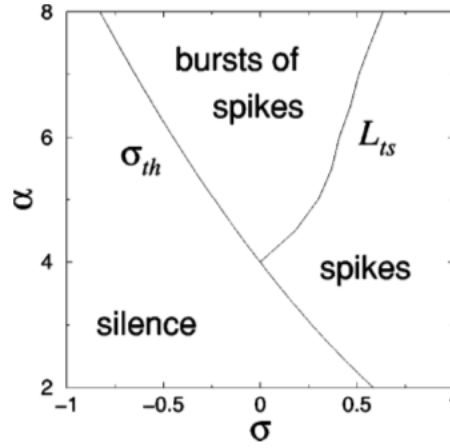


Figura A.2: Diagrama de bifurcación de la actividad del mapa de Rulkov en el plano  $(\mu, \alpha)$ . Figura reproducida de (Rulkov, 2002)

## A.4. Modelos de sinapsis

### A.4.1. Sinapsis eléctrica

La sinapsis eléctrica se define como la diferencia de los potenciales de ambas neuronas multiplicada por una conductancia que representa la aceptación que muestra la neurona receptora a la señal externa.

$$I_{ab} = g_{ab}(V_a - V_b)$$

### A.4.2. Sinapsis química gradual

Se ha utilizado un modelo de sinapsis química (Golowasch et al., 1999) que permite simular las dinámicas lentas y rápidas del proceso. Para ello cuenta con dos posibles corrientes. Pueden ser utilizadas por separado o ser sumadas dependiendo del experimento que se desee realizar.

#### Rápida

$$I_{fastX} = \sum_Y \frac{g_{fastYX}(V_X - E_{syn})}{1 + \exp(s_{fast}(V_{fast} + V_Y))}$$

## Lenta

$$I_{slowX} = \sum_Y g_{slowYX} m_{slowX} (V_x - E_{syn})$$

$$\frac{dm_{slowX}}{dt} = \frac{k_1 X (1 - m_{slowX})}{1 + \exp(s_{slow}(V_{slow} - V_{AB}))} - k_2 X m_{slowX}$$

### A.4.3. Sinapsis química dinámica

También puede utilizarse una sinapsis química que caracterice los recursos, neurotransmisores, en función de la cantidad disponible (Destexhe et al., 1994).

Para ello se utiliza una primera ecuación que representa el resultado de la sinapsis en la neurona postsináptica mientras otras tres caracterizan los recursos siendo, E recursos efectivos, I inactivos y R recuperados.

$$EPSC_{n+1} = EPSC_n (1 - U_{SE}) e^{-\Delta t / \tau_{rec}} + A_{SE} U_{SE} (1 - e^{-\Delta / \tau_{rec}})$$

$$\frac{dR}{dt} = \frac{I}{\tau_{rec}} - U_{SE} R \delta(t - t_{AP})$$

$$\frac{dE}{dt} = -\frac{E}{\tau_{inact}} - U_{SE} R \delta(t - t_{AP})$$

$$I = 1 - R - E$$



## Clamp-cli

Los algoritmos desarrollados en este trabajo se encuentran implementados en un programa llamado Clamp-CLI, realizado conjuntamente con otro Trabajo Fin de Máster (Amaducci, 2017).

Este otro trabajo se centra en la construcción de una librería de modelos neuronales. El diseño modular de este programa permite añadir modelos y realizar de manera sencilla los circuitos híbridos. Este programa requiere de los algoritmos expuestos en este TFM e implementa el protocolo diseñado.

En este anexo se incluye información necesaria para su utilización y comprensión. El código y las instrucciones de este anexo se encuentran disponibles en su última versión en:

- [www.github.com/manurs/clamp-cli](https://www.github.com/manurs/clamp-cli)

Debe tenerse en cuenta que el presente anexo puede quedarse desfasado en un futuro ante la modificación del programa, la versión de las presentes instrucciones se encuentra disponible en:

- [www.github.com/manurs/clamp-cli/tree/817660eeca697f792984b9a8fb6130028b072741](https://www.github.com/manurs/clamp-cli/tree/817660eeca697f792984b9a8fb6130028b072741)
- [www.github.com/manurs/clamp-cli/releases/tag/v0.1](https://www.github.com/manurs/clamp-cli/releases/tag/v0.1)

### B.1. Organización del proyecto

---

Atendiendo a las funciones que dirigen el flujo del programa nos encontramos con los siguientes componentes:

- **main:** es el punto de entrada al programa cuando se ejecuta y se encarga de establecer el experimento de acuerdo a los parámetros de llamada del programa, determinar los archivos en los que guardar la información y crear los dos hilos de ejecución de programa, rellenando las estructuras de comunicación utilizadas.
- **writer\_thread:** hilo creado por el programa que se encarga de recibir mensajes desde **rt\_thread** para almacenar la información en ficheros.

- **rt\_thread:** hilo que se ejecuta en tiempo real. En este hilo se interactúa con la neurona viva y contiene el flujo de funcionamiento del protocolo. Cuenta además con una función auxiliar para convertirse en un hilo en tiempo real.

Adicionalmente estos componentes, en especial **rt\_thread**, utilizan librerías desarrolladas para abstraer el uso de ciertas herramientas necesarias:

- **time\_management:** funciones para manejar las estructuras de tiempo y realizar operaciones sobre ellas.
- **comedi\_functions:** funciones para la utilización de las librerías Comedi y comunicarse a través de ellas con la tarjeta de adquisición de datos.
- **model\_library:** implementación de los modelos de neuronas y de sinapsis.
- **calibrate\_functions\_phase1:** funciones de calibración y de establecimiento de la conexión híbrida para el primer subsistema (ver secciones 4.2 y 5.2).
- **calibrate\_functions\_phase2:** funciones de calibración y de establecimiento de la conexión híbrida para el segundo subsistema (ver secciones 4.2 y 5.3).

---

## B.2. Instalación

Si se cuenta con un kernel compilado con `PREEMPT_RT` para conseguir tiempo real y con Comedi para manejar la tarjeta de adquisición de datos es posible compilar el programa mediante la instrucción **make** y proceder a su utilización.

### B.2.1. Guía para Debian

Se ofrece una guía para obtener un sistema operativo con un kernel válido y las librerías Comedi. En concreto esta guía está desarrollada sobre la versión 9.0 de Debian.

1. Descargar Debian de [www.debian.org](http://www.debian.org)
2. Instalar Debian:
  - Cuenta con interfaz de instalación gráfica, sin embargo puede ser necesaria asistencia a la hora de particionar el disco especialmente si se desea conservar un sistema operativo anterior como Windows.
  - Como entorno gráfico puede ser seleccionado cualquiera de los disponibles. En máquinas con pocos recursos se recomiendan opciones ligeras. Si se selecciona Xfce puede configurarse siguiendo las instrucciones de [www.github.com/manurs/xfce\\_config/tree/master/config\\_files/debian\\_basic](http://www.github.com/manurs/xfce_config/tree/master/config_files/debian_basic).
3. Configuración del comando `sudo` (necesario para la instalación posterior):  
[www.github.com/manurs/guides.md/blob/master/sudo.md](http://www.github.com/manurs/guides.md/blob/master/sudo.md)
4. Comandos de terminal para la instalación del kernel con tiempo real:

```
$ sudo apt install linux-image-rt-amd64
```

Código B.1: Comando para la instalación de Linux `PREEMPT_RT`



## 5. Comandos de terminal para la instalación de las librerías Comedi:

```
$ sudo apt install git build-essential dkms automake linux-headers-rt-amd64
$ git clone https://github.com/Linux-Comedi/comedi.git
$ ./comedi/autogen.sh
$ sudo dkms add ./comedi
$ sudo dkms install comedi/0.7.76.1+20170502git -1.nodist
$ sudo depmod -a
$ sudo apt install libcomedi0 libcomedi-dev
```

Código B.2: Comandos para la compilación de librerías Comedi

## 6. Descarga del programa e instalación:

```
$ git clone https://github.com/manurs/clamp-cli.git
$ cd clamp-cli
$ make
```

Código B.3: Comandos para descargar e instalar Clamp-cli

### B.3. Utilización

El programa define su comportamiento de acuerdo a los argumentos de llamada al programa. Si ejecutamos el programa con el argumento **-h** es posible obtener información sobre los posibles argumentos:

```
-f, --frequency: sample frequency (in KHz)
-t, --time: simulation time (in s)
-m, --model: neural model (0 = Izhikevich, 1 = Hindmarsh-Rose, 2 = Rulkov Map)
-s, --synapse: synapse type (0 = electrical, 1 = gradual)
-a, --antiphase: turn on antiphase
-c, --calibration: automatic calibration process
  - Don't use with antiphase
  - Synapse will be ignored
  - Number codes in github wiki
-I, --Important: mark experiment in summary.txt
-i, --input_channels: input channels, separated by commas (ej: 0,2,3,7)
-o, --output_channels: output channels, separated by commas (ej: 0,2,3,7)
-h, --help: print this help
```

Código B.4: Guía de argumentos de llamada de Clamp-cli

```
1 - Electrical Synapse (phase) - ECM percentage
2 - Electrical Synapse (phase) - ECM slope
3 - Electrical Synapse (phase) - ECM var
4 - Electrical Synapse (phase) - Phase var
5 - Electrical Synapse (antiphase) - Phase var
6 - Map HR mu (with electrical synapse in phase)
7 - Map g (Chemical Synapse)
8 - Map k1 and k2 (Chemical Synapse)
```

Código B.5: Guía de códigos de la opción de autocalibración -c de Clamp-cli

## B.4. Fichero de datos

Los resultados de las ejecuciones se guardan en la carpeta data. Los ficheros con información a su vez son almacenados en subcarpetas agrupados por días. Es decir que los ficheros siguen el siguiente patrón:

`data/****y**m**d/**h**m**s_tipo.txt`

En cada carpeta se genera además un fichero resumen (**summary.txt**) que indica los detalles de cada experimento (fecha, frecuencia de observación, modelo utilizado...).

### B.4.1. Fichero 1

La primera fila de este archivo incluye dos columnas que indican respectivamente cuantos canales de entrada y cuantos de salida han sido grabados.

A partir de la segunda línea se incluyen para cada instante las siguientes columnas:

- Tiempo unix
- Tiempo desde el inicio de experimento
- Número de interacción
- Latencia
- Voltaje del modelo
- Voltaje del modelo escalado
- Corriente del modelo a la tarjeta
- Corriente introducida en el modelo
- Primer canal de entrada
- ...
- Último canal de entrada
- Primer canal de salida
- ...
- Último canal de salida

Ejemplo del fichero:

```
1 2
118151610.862160 5140.804176 0 4176 -71.047950 -4.942058 -0.016286 -0.088902
-4.835431 -0.016286 -4.942058
118151610.962176 5140.904192 1 4192 -71.045875 -4.942015 -0.016287 -0.087665
-4.848554 -0.016287 -4.942015
118151611.062198 5141.004214 2 4214 -71.043591 -4.941968 -0.016288 -0.086715
-4.858625 -0.016288 -4.941968
118151611.162169 5141.104185 3 4185 -71.041693 -4.941929 -0.016289 -0.088557
-4.839094 -0.016289 -4.941929
118151611.262201 5141.204217 4 4217 -71.039982 -4.941894 -0.016289 -0.089479
-4.829328 -0.016289 -4.941894
```

|                  |             |           |      |            |           |           |           |
|------------------|-------------|-----------|------|------------|-----------|-----------|-----------|
| 118151611.362173 | 5141.304189 | 5         | 4189 | -71.037881 | -4.941851 | -0.016290 | -0.087694 |
| -4.848249        | -0.016290   | -4.941851 |      |            |           |           |           |
| 118151611.462222 | 5141.404238 | 6         | 4238 | -71.035799 | -4.941808 | -0.016291 | -0.087809 |
| -4.847028        | -0.016291   | -4.941808 |      |            |           |           |           |
| 118151611.562170 | 5141.504186 | 7         | 4186 | -71.034052 | -4.941772 | -0.016292 | -0.089422 |
| -4.829938        | -0.016292   | -4.941772 |      |            |           |           |           |
| 118151611.662193 | 5141.604209 | 51414     | 4209 | -71.032448 | -4.941739 | -0.016292 | -0.090141 |
| -4.822309        | -0.016292   | -4.941739 |      |            |           |           |           |

### B.4.2. Fichero 2

El segundo fichero incluye información respectiva al ajuste dinámico de parámetros:

- Tiempo desde el inicio de experimento
- Número de interacción
- ECM/Otro
- Varianza/Otro
- Primera conductancia
- ...
- Última conductancia

Ejemplo del fichero:

|             |       |          |          |          |          |          |          |
|-------------|-------|----------|----------|----------|----------|----------|----------|
| 5140.804176 | 51406 | 0.400000 | 0.010000 | 0.200000 | 0.000000 | 0.000000 | 0.020000 |
| 5140.904192 | 51407 | 0.400000 | 0.010000 | 0.200000 | 0.000000 | 0.000000 | 0.020000 |
| 5141.004214 | 51408 | 0.400000 | 0.010000 | 0.200000 | 0.000000 | 0.000000 | 0.020000 |
| 5141.104185 | 51409 | 0.400000 | 0.010000 | 0.200000 | 0.000000 | 0.000000 | 0.020000 |
| 5141.204217 | 51410 | 0.400000 | 0.010000 | 0.200000 | 0.000000 | 0.000000 | 0.020000 |
| 5141.304189 | 51411 | 0.400000 | 0.010000 | 0.200000 | 0.000000 | 0.000000 | 0.020000 |
| 5141.404238 | 51412 | 0.400000 | 0.010000 | 0.200000 | 0.000000 | 0.000000 | 0.020000 |
| 5141.504186 | 51413 | 0.400000 | 0.010000 | 0.200000 | 0.000000 | 0.000000 | 0.020000 |
| 5141.604209 | 51414 | 0.400000 | 0.010000 | 0.200000 | 0.000000 | 0.000000 | 0.020000 |

## B.5. Representación gráfica

El programa incluye códigos Python para visualizar los datos con la librería Matplotlib. Estos programas se encargan de leer de manera adecuada los ficheros de datos, almacenando la información en variable de Python. Estos programas pueden ser adaptados para seleccionar que información será representada.





## Estudio de latencias

Para comprobar el funcionamiento de la tecnología de tiempo real se han realizado pruebas con el objetivo de medir las latencias de clamp-cli. Para ello se han cargado todos los procesadores disponibles y se ha realizado un experimento tanto con el kernel de tiempo real como desactivado. La herramienta utilizada para generar carga computacional ha sido sysbench ([www.github.com/akopytov/sysbench](http://www.github.com/akopytov/sysbench)).

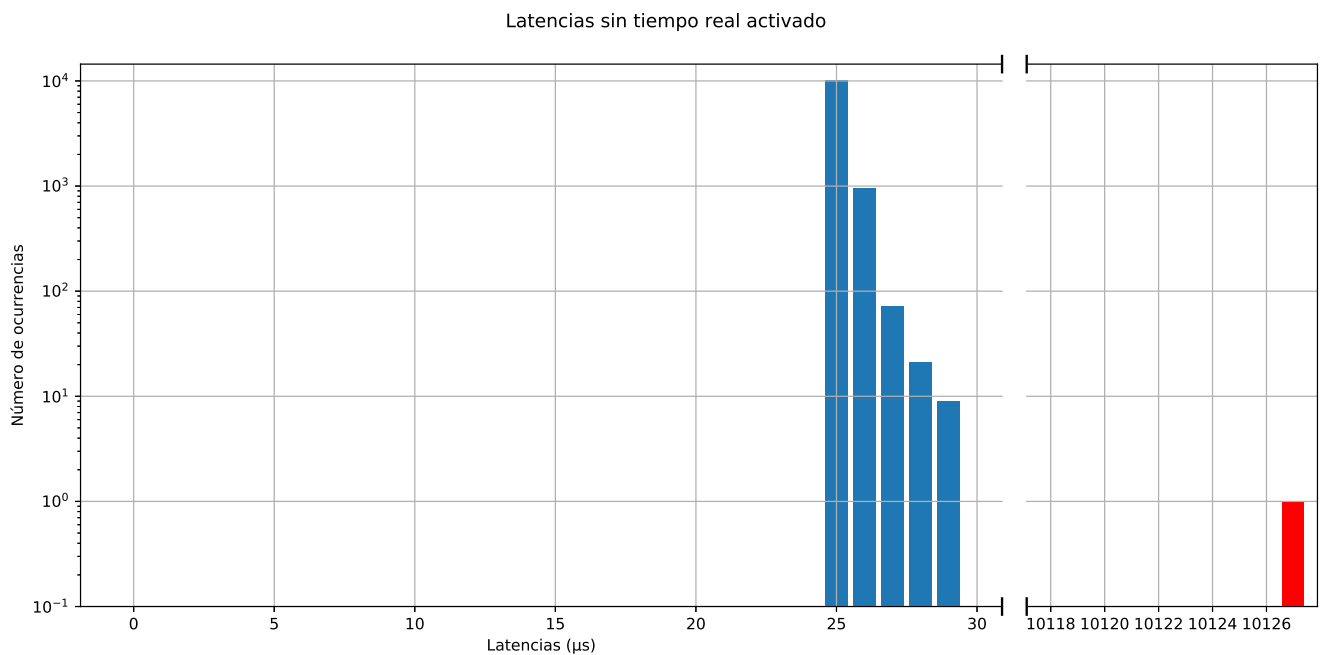


Figura C.1: Latencias sin kernel de tiempo real.

En la figura C.1 se puede observar como los resultados no son válidos sin un sistema operativo con tiempo real. La latencia mayor ha sido de  $10.000 \mu s$  cuando el periodo de interacción es de 100 a  $50 \mu s$ . En cambio, en la figura C.2 realizada con Linux PREEMPT\_RT los peores resultados son mejores que las menores latencias conseguidas en la primera figura.

Cabe destacar la diferencia de escalas de ambas figuras. Mientras que con el tiempo real desactivado los resultados positivos, aquellos por debajo de los  $30 \mu s$ , son mínimos y la mayor parte se encuentran por encima (no se muestran en la figura por motivos de escala) con el tiempo real activado todas las latencias están por debajo de los  $18 \mu s$ .

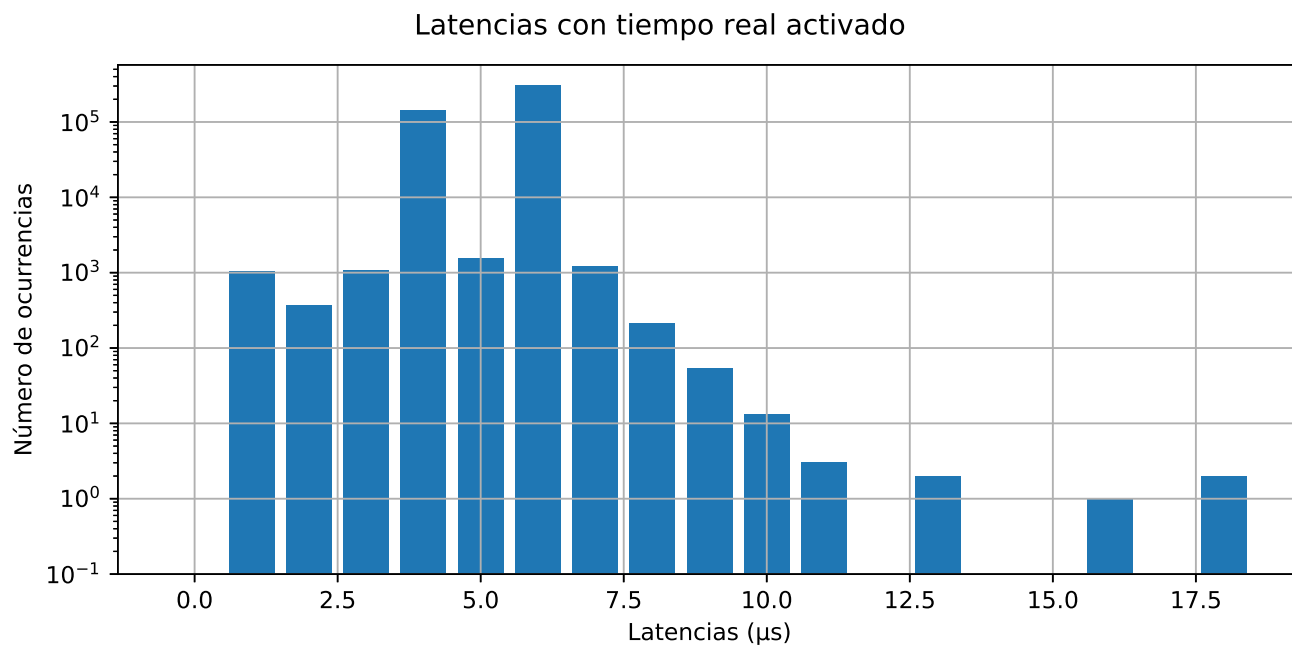


Figura C.2: Latencias con kernel de tiempo real.



## Difusión del trabajo

El trabajo realizado será presentado en los siguientes congresos:

- M. Reyes-Sánchez, I. Elices, R. Amaducci, C. Muñoz, F. B. Rodríguez, P. Varona. Automatic calibration for hybrid circuits of living and artificial neurons. Aceptado para Computational Neuroscience Meeting CNS 2017 Antwerp.
- R. Amaducci, C. Muñoz, M. Reyes-Sánchez, F. B. Rodríguez, P. Varona. On the need for real time software technology in closed-loop neuroscience. Aceptado para Computational Neuroscience Meeting CNS 2017 Antwerp.
- M. Reyes-Sánchez, I. Elices, R. Amaducci, C. Muñoz, F. B. Rodríguez, P. Varona. Dynamic adaptation for hybrid circuit interactions. Aceptado para congreso SENC 2017 Alicante.
- R. Amaducci, C. Muñoz, M. Reyes-Sánchez, F. B. Rodríguez, P. Varona. Standardized real-time software in experimental neuroscience. Aceptado para congreso SENC 2017 Alicante.

